

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-067177  
(43)Date of publication of application : 07.03.2003

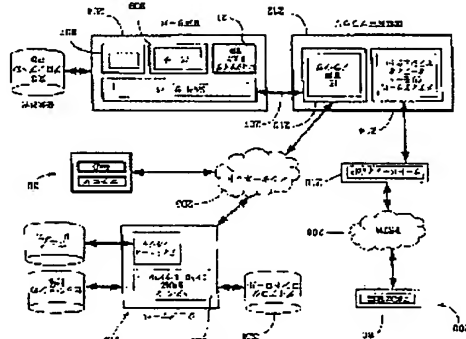
(51)Int.Cl.  
G06F 3/16  
G06F 13/00  
G06F 15/00  
G06F 15/02

(21)Application number : 2002-132053 (71)Applicant : MICROSOFT CORP  
(22)Date of filing : 07.05.2002 (72)Inventor : WANG KUANSAN  
HON HSIAO-WUEN

(30)Priority  
Priority number : 2001 289041 Priority date : 04.05.2001 Priority country : US  
2001 960232 20.09.2001 US  
2002 117141 05.04.2002 US

## (54) SYSTEM AND METHOD HAVING WEB CORRESPONDENCE RECOGNITION ARCHITECTURE

(57)Abstract:  
PROBLEM TO BE SOLVED: To provide a system having a Web correspondence recognition architecture used in presenting recognition of input with a universal architecture.  
SOLUTION: A server/client system includes a network having a Web server 202 having information accessible from remote. A client device includes a microphone, and a rendering component such as a speaker or a display. The client device is constructed to obtain information from the Web server 202, records input data associated with a field contained in the information, and is adapted to designate syntax used in recognition and transmit the input data to a remote position. A recognition server 204 receives the input data and designation of syntax, and returns the data showing what is recognized at least one of the client and Web server.



## LEGAL STATUS

[Date of request for examination] 06.05.2005  
[Date of sending the examiner's decision of rejection]  
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]  
[Date of final disposal for application]

(19)日本国特許庁 (JP) (12)公開特許公報 (A)

(11)特許公開公報番号  
特開2003-67177  
(P2003-67177A)

(43)公開日 平成15年3月7日(2003.3.7)

(51)Int.Cl.	識別記号	FI	特許公開公報番号
G06F 3/16	320	G06F 3/16	320H 5B019
13/00	510	13/00	510C 5B085
15/00	310	15/00	310B
15/02	310	15/02	310D
	325		325B

審査請求 未請求 請求項の数24 OI (全49頁) 最終頁に続く

(21)出願番号	特開2002-132053(P2002-132053)	(71)出願人	391055833 マイクロソフト コーポレーション MICROSOFT CORPORATI ON アメリカ合衆国 ワシントン州 98052- 6399 レッドモンド ワン マイクロソフ ト ウェイ (番地なし) 100077481 (74)代理人 井理士 谷 鶴一 (外2名)
(22)出願日	平成14年5月7日(2002.5.7)		
(31)優先権主張番号	6,0/2,89,041		
(32)優先日	平成13年5月4日(2001.5.4)		
(33)優先権主張国	米国 (US)		
(31)優先権主張番号	09/960,232		
(32)優先日	平成13年9月20日(2001.9.20)		
(33)優先権主張国	米国 (US)		
(31)優先権主張番号	10/117,141		
(32)優先日	平成14年4月5日(2002.4.5)		
(33)優先権主張国	米国 (US)		

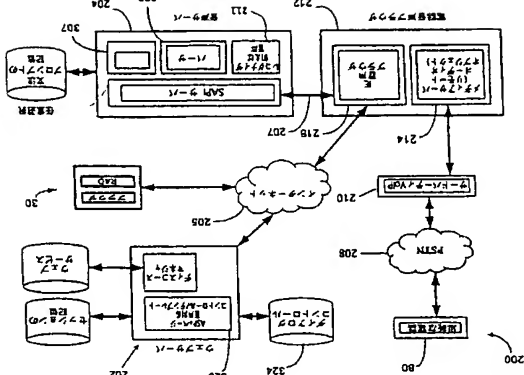
最終頁に続く

(54)【発明の名称】 ウェブ対応認識アーキテクチャを有するシステムおよびその方法

(57)【要約】

【課題】 入力認識を提供するために使用されるウェブ対応認識アーキテクチャを有するシステムに、統一したアーキテクチャを持たせる。

【解決手段】 サーバ/クライアントシステムは、リモートからアクセスすることのできる情報を含むウェブサーバ202を有するネットワークを含む。クライアントデバイス204は、マイクロフォンと、スピーカまたはディスプレイなどのレンダリング構成要素を含む。クライアントデバイスは、ウェブサーバ202から情報入手し、その情報中に含まれる、フィールドと関連付けられ、入力データを記録するように構成し、また、認識に使用する文法の指示とともに、入力データを遠隔位置に送信するように適合する。認識サーバ204は、入力データおよび文法の指示を受信し、また、何が認識されたかを指示するデータを、クライアントおよびウェブサーバの少なくとも1つに戻す。



(2) 特開2003-67177 2

【特許請求の範囲】  
【請求項1】 データを処理するサーバ/クライアントシステムであって、  
リモートでアクセスすることのできる情報を有するウェブサーバと、  
入力すべきフィールドを指示するレンダリングデバイスを有するクライアントデバイスであって、ユーザがその後行入力の対象とするフィールドを指示すると、前記フィールドの各々と関連付けられた入力データを記録するように構成され、かつ前記入力データを遠隔位置に送信するように適合されたクライアントデバイスと、  
前記入力データを受領し、何が認識されたかを表すデータを、前記クライアントおよび前記ウェブサーバの少なくとも1つに原すように構成された認識サーバを含むネットワークを備えたことを特徴とするシステム、  
【請求項2】 前記ウェブサーバから受領し、前記クライアントデバイスに提供する前記情報は、マークアップ言語であることを特徴とする請求項1に記載のシステム、  
【請求項3】 前記クライアントデバイスが受領する前記マークアップ言語は、1つまたはいくつかのマークアップ部分、および1つまたはいくつかのスク립タ部分を含むことを特徴とする請求項2に記載のシステム、  
【請求項4】 前記マークアップ言語は、文法をフィールドに関連付ける指示を含むことを特徴とする請求項3に記載のシステム、  
【請求項5】 前記認識サーバは、前記入力データおよび前記文法の指示を受領することを特徴とする請求項4に記載のシステム、  
【請求項6】 前記入力データを前記リモートサーバに送信する前に前記入力データを正規化する上に、前記クライアントが適合されることを特徴とする請求項1に記載のシステム、  
【請求項7】 前記ウェブサーバは、前記クライアントデバイスのためにマークアップ言語を動的に生成するサーバ/クライアントデバイスを含むことを特徴とする請求項2に記載のシステム、  
【請求項8】 マイクフォンおよびスピーカを有する第2のクライアントデバイスをさらに含み、該第2のクライアントデバイスは、前記ユーザに与えるプロンプトに応じて、各フィールドセットに関連付けられた音声データを記録するように構成され、かつ、前記音声データを前記認識サーバに送信するように適合されることを特徴とする請求項1に記載のシステム、  
【請求項9】 前記第2のクライアントデバイスは、前記ウェブサーバからのコンテningをレンダリングすることのできる電話機および音声ブラウザを含むことを特徴とする請求項8に記載のシステム、  
【請求項10】 前記ウェブサーバから受領し、前記クライアントデバイスの各々に提供する前記情報は、マー

(3) 特開2003-67177 3

【請求項20】 前記クライアントおよび前記認識サーバを単一のマシンに配置することを特徴とする請求項16に記載のシステム、  
【請求項21】 クライアント/サーバシステムで音声認識を処理する方法であって、  
前記クライアントデバイスのユーザから入力データを手すように構成されたエクステンションを有するマークアップ言語ページを、ウェブサーバからクライアントデバイスに送信するステップと、  
前記マークアップ言語ページを前記クライアントデバイスでレンダリングするステップと、  
前記ユーザからの入力に応じて、入力データを入力するステップと、  
前記入力データおよび関連付けられた文法の指示を、前記クライアントデバイスから遠隔に位置する認識サーバに送信するステップと、  
前記クライアントデバイスおよびウェブサーバの少なくとも1つで、何が入力されたかを表す認識結果を、前記認識サーバから受領するステップとを備えたことを特徴とする方法、  
【請求項22】 前記マークアップ言語をレンダリングするステップは、データ入力のフィールドを表示するステップを含み、入力データを入力するステップは、その後行入力をどのフィールドに関連付けるかについて指示を前記ユーザから受領するステップを含むことを特徴とする請求項21に記載の方法、  
【請求項23】 前記マークアップ言語をレンダリングするステップは、前記ユーザに対して音声により指示を促すステップを含むことを特徴とする請求項21に記載の方法、  
【請求項24】 前記マークアップ言語はスク립タ言語を原稿することとを特徴とする請求項21に記載の方法、  
【発明の詳細な説明】  
【0001】  
【発明の属する技術分野】 本発明は、インターネットなどのワイアレスネットワークを介した情報のアクセスに関する。より詳細には、本発明は、各種の方法を使用してクライアント側で情報およびコントロールを入力することを可能にするウェブ対応認識に関する。  
【0002】  
【従来の技術】 人々が、個人情報マネージャ (PIM)、デバイス、および携帯電話のような小型のコンピュータでインデックスを日常活動で使用する頻度は増す方である。現在では、こうしたデバイスを作動させるのに使われるマイクロプロセッサに利用できる処理能力が増大したことにより、これらデバイスの機能性が高まっている。例えば現在、携帯電話の多くは、アドレス、電話番号などの個人情報、携帯電話の多くは、アドレス、電話番号などの個人情報に記憶して使用できるだけでなく、インターネットの

(3) 特開2003-67177 4

アクセスおよびブラウザにも使用することができる。  
【0003】 こうしたコンピュータ化デバイスをインターネットブラウザに使用し、あるいは他のサーバ/クライアントアプリケーションで使用するところから、情報をコンピュータ化デバイスに入力することが必要となる。不都合なのは、操作を容易にするためにこうしたデバイスは可能な限り小さくしたいという要求があり、利用可能なコンピュータ化デバイスの表面積が限られているために、アルファベットの全文字を個別のボタンとして備える従来のキーボードが例は不可能であることである。  
【0004】 最近、VoiceXML (音声拡張可能マークアップ言語) の使用によるなどの音声ポータルが進歩し、電話だけを使用してインターネットコンテンツにアクセスすることが可能になっている。このアーキテクチャでは、ドキュメントサーバ (例えばウェブサーバ) が、VoiceXMLインテプブリタを通じてクライアントからの要求を処理する。ウェブサーバはそれに応答してVoiceXMLドキュメントを生成することであり、このドキュメントはVoiceXMLインテプブリタによって処理し、ユーザに対して音声としてレンダリングされる。ユーザは、音声認識を通じて音声コマンドを使用することにより、ウェブをナビゲートすることができる。  
【0005】 VoiceXMLは、フロー制御タグを用いるマークアップ言語であるが、フロー制御は、イベントing (eventing) および個別のスク립タを含むHTML (ハイパーテキストマークアップ言語) のフロー制御モデルには従わない。VoiceXMLは一般に、電話ベースの音声のみの対話に特に適したフォーマットアルゴリズムを含むが、このシステムでは通常、ユーザから得られる情報をシステマまたはアプリケーションによって制御する。グラフィカルユーザインタフェースも提供し、クライアントサーバ間で利用することのできるアプリケーションにVoiceXMLを直接組み込むには、開発者は、2つの形態のウェブサービスを用意する必要がある。すなわち、VoiceXMLのオーサリングと、HTML (など) を使用したオーサリングであるが、これらはそれぞれ異なるフロー制御モデルに従っている。  
【0006】  
【発明が解決しようとする課題】 したがって、インターネットなどのサーバ/クライアントアーキテクチャで音声認識を提供するのに使用されるアーキテクチャ、またはその部分、および方法に改良を加えることが現在必要とされている。音声認識用のオーサリングツールは、PIM、電話などの小型のコンピュータ化デバイスに容易に適合できなければならない。前述の不利点の1つ、いくつかは、またはすべてに対処するウェブサービスのアークテクチャまたは方法が特に必要とされる。

他の入力方法が可能なアーキテクチャも有益であろう。  
【0007】本発明は、このような課題に鑑みてなされたもので、その目的とするところは、インターネットなどのサーバ/クライアントアーキテクチャで入力の認識を提供するのに使用される、統一したアーキテクチャを備えたウェブ対応認識アーキテクチャを有するシステムおよびその方法を提供することにある。

【0008】  
【課題を解決するための手段】データ処理用のサーバ/クライアントシステムは、リモートにアクセスできる情報を含んだウェブサーバを有するネットワークを含む。クライアントデバイスは、マイクروفフォンと、スピーカまたはディスプレイなどのレンダリング構成要素を含む。クライアントデバイスは、ウェブサーバから情報を受取り、その情報に含まれるフィールドと関連付けられた入力データを記録するように構成する。クライアントデバイスは、認識に使用する文法の指示とともに入力データを通信回線に送信するように適合する。認識サーバは、何が入力されたかを示すデータをクライアントおよびウェブサーバの少なくとも1つに伝す。

【0009】  
【発明の実施の形態】ウェブベース認識のアーキテクチャおよびその実施方法を説明する前に、このアーキテクチャで機能する方法が可能なコンピュータインテグレーションについて全般的に説明しておくことが有用であろう。本明細書で図1を参照すると、下記で論じるこの他のコンピュータM、PDAなどの例示的形態が30に表されている。ただし、本発明は、下記で論じるこの他のコンピュータに面積に限られたコンピュータインテグレーションでは表裏が逆であることも企図している。例えば、電話およびインターネットはデータ管理デバイスも、本発明から利益を受けることができる。このようなデバイスは、既存の携帯個人情報管理デバイスおよびその他の携帯電子デバイスと比較して高いユーティリティを備え、そのデバイスの機能とコンピュータサイズにより、ユーザがデバイスを常に携帯することとを低く思われる。したがって、本明細書に記載するアーキテクチャの範囲は、本明細書に記載する例示的なデータ管理デバイスまたはPIMデバイス、電話機、またはコンピュータの開示によつては制限しないものとする。

【0010】データ管理モバイルデバイス30の例示的な形態を図1に示す。モバイルデバイス30は筐体32を含み、ディスプレイ34を含むユーザインタフェースを有する。ユーザインタフェースには、スタイルラス33と合わせ、接触感知式の表示面を使用する。スタイルラス33は、指定された座標でディスプレイ34を押す、またはディスプレイ34に接触して、フィールドを選択し、カーソルの開始位置を選択的に移動するのに使用

(音声、DTMF、手書き、ジェスチャ、またはコンピュータ画像)を含む。一例として、デバイス30のユーザからの音声による情報、命令、またはコマンドにตอบสนองして、マイクروفフォン29が音声信号を提供し、それをA/D変換器37でデジタル化する。音声認識プログラムは、デジタル化した音声信号に正規化および/または特徴抽出機能を行って、中間の音声認識結果を得る。無線トランシーバ52または通信インタフェース60を使用して、下記で説明し、図5のアーキテクチャに示すモータの認識サーバ20に音声データを送信する。その後認識結果をモバイルデバイス30に戻して、そこでレンダリング(例えば視覚的かつ/または可聴的に)を行い、最終的にウェブサーバ202(図5)に送信するが、本明細書でウェブサーバ202とモバイルデバイス30はクライアント/サーバ関係で動作している。これと同様の処理を、他の形態の入力にも使用することができ、例えば、手書き入力、デバイス30での前処理により、または前処理によらずにデジタル化することができ、音声データと同様に、この形態の入力も認識のために認識サーバ204に送信することができ、認識結果が、デバイス30および/またはウェブサーバ202の少なくともどちらかに戻される。同様に、DTMFデータ、ジェスチャデータ、および視覚データも同じように処理することができ、入力形態に応じて、デバイス30(および下記で説明する他の形態のクライアント)は、カメラや視覚入力が必要なハードウェアを含む。

【0015】図3は、携帯電話80の一側面示実施形態の平面図である。電話機80は、ディスプレイ82およびキーパッド84を含む。一般に、図2のブロック図は図3の電話機にも該当するが、他機能を行うために必須の追加回路が必要になることもある。例えば、図3の実施形態には、電話としての動作に必要なトランシーバが必要になるが、このような回路は本発明には関連しない。

【0016】上記の携帯式またはモバイル型のコンピュータインテグレーション以外にも、本発明は、一般的なデスクトップコンピュータなど数多くの他のコンピュータインテグレーションにも使用できることが理解されよう。例えば、身体能力が限られたユーザにとって完全な英数字キーボードなど他の従来型の入力装置の操作が困難である場合には、本発明は、そのようなユーザがコンピュータまたは他のコンピュータインテグレーションにアクセスを入力を可能にする。

【0017】本発明はまた、数多くの他の汎用または特殊目的のコンピュータインテグレーション、環境、または構成での動作が可能である。本発明とともに使用するのに適した周知のコンピュータインテグレーション、環境、および/または構成の例には、従来型の(regular)電話(画面を備えない)、パーソナルコンピュータ、サーバコンピュータ、携帯用デバイスまたはラップトップ

デバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラム可能な家庭用電化製品、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、上記のシステムまたはデバイスなどのうち任意のものを含む分散型コンピュータインテグレーション環境が含まれるがこれらに限定するものではない。

【0018】以下は、図4に示す汎用コンピュータ120の簡単な説明である。ただし、この場合もコンピュータ120は、適切なコンピュータインテグレーションの一例に過ぎず、本発明の使用または機能性の範囲に關して何らの制限を示唆するものではない。また、コンピュータ120は、この図に示す構成要素のいずれか、またはその組合せに関する依存性や要件を有するものとも解釈すべきではない。

【0019】本発明は、プログラムモジュールなどコンピュータで実行するコンピュータ実行可能命令の一般的な状況で説明することができる。一般に、プログラムモジュールには、特定タスクを実行する、または特定の抽象データタイプを実施する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。本発明はまた、通信ネットワークを通じてリンクした遠隔処理装置によってタスクを実行する分散型コンピュータインテグレーション環境でも実施することができ、分散型コンピュータインテグレーションでは、プログラムモジュールは、メモリ記憶装置を含む、ローカルおよびリモートどちらのコンピュータ記憶媒体に置いていてもよい。以下、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセス実行可能命令として実施することができ、この命令はこの形態のコンピュータ読み取り可能な記録媒体にも書き込むことができる。

【0020】図4を参照すると、コンピュータ120の構成要素には、プロセッサ140、システムメモリ150、およびシステムメモリを含む各種システム構成要素をプロセッサ140に結合するシステムバス141が含まれるが、これらに限定しない。システムバス141は、メモリまたはメモリコントローラ、周辺バス、および各種バスアーキテクチャのうち任意のものを使用したローカルバスを含む各種バス構造のうち任意のも

のよい。このようなアーキテクチャには、例えば、ISA(Industry Standard Architecture)バス、USB(Universal Serial Bus)、MCA(Micro Channel Architecture)バス、EISA(Enhanced ISA)バス、VESA(Video Electronics Standards Association)ローカルバス、およびメニバスとしても知られるPCI(Peripheral Component Interconnect)バ

可能な記録媒体も含むことができる。図4には、取外し不能、不揮発性の磁気媒体との読み出しまたは書き込みを行うハードディスクドライブ161、リムーバブル、不揮発性の磁気ディスク172との読み出しまたは書き込みを行う磁気ディスクドライブ171、およびCD、R、OMや他の光媒体などのリムーバブル、不揮発性の光ディスク176との読み出しまたは書き込みを行う光ディスクドライブ175を示すが、これらは例にすぎない。この例示的動作環境で使用できる、その他のリムーバブル/取外し不能、揮発性/不揮発性のコンピュータ記録媒体には、磁気テープカセット、フラッシュメモリーカード、デジタル多用ディスク、デジタルビデオテープ、ソリッドステートRAM、ソリッドステートROMなどが含まれるがこれらに限定しない。ハードディスクドライブ161は、通例、インタフェース170など取外し不能のメモリーインタフェースを通じてシステムバス141に接続し、磁気ディスクドライブ171および光ディスクドライブ175は通例、インタフェース170などのリムーバブルメモリーインタフェースによってシステムバス141に接続する。

【0024】上記で説明し、図4に示すドライブおよびそれに関連するコンピュータ記録媒体は、コンピュータ120のコンピュータ可読命令、データ構造、プログラムモジュール、およびその他のデータの記憶を提供する。例えば、図4では、ハードディスクドライブ161は、オペレーティングシステム164、アプリケーション166、およびプログラムモジュール165、他のプログラムモジュール156、およびプログラムデータ157と同じものでも、異なるものでもよいことに留意されたい。本明細書では、オペレーティングシステム164、アプリケーション166、およびプログラムモジュール165、他のプログラムモジュール156、およびプログラムデータ167が少なくとも異なる番号を与えている。

【0025】ユーザは、キーボード182、マイクロプロセッサ183、およびマウスやトラックボール、タッチパッドなどのポインティングデバイス181などの入力装置を通じて、コンピュータ120にコマンドおよび情報を入力することができる。この他の入力装置（図示せず）には、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナなどが含まれる。これらの入力装置およびその他の入力装置は、多くの場合、システムバスに結合したユーザ入力インタフェース180を通じてプロセッサ140に接続するが、パラレルポート、ゲームポート、あるいはユニバーサルシリアルバス（USB）など他のインタフェースおよびバス構造によって接続することもできる。モニタ184または他の表示装置

置も、ビデオインタフェース185などのインタフェースを紹介し、システムバス141に接続する。コンピュータは、モニタ以外にも、スピーカ187およびプリンタ186など他の周辺出力装置も含むことができ、これらは出力周辺インタフェース188を通じて接続することができる。

【0026】コンピュータ120は、リモートコンピュータ194など1つまたは複数のリモートコンピュータへの通信接続を使用するネットワーク環境で動作することができる。リモートコンピュータ194は、パーソナルコンピュータ、携帯用デバイス、サーバ、ルータ、ネットワークPC、ピアデバイス、または他の一般的なネットワークノードでよく、通例は上記でコンピュータ120との関連で説明した要素の多くまたはすべてを含む。図4に示す論理環境には、ローカルエリアネットワーク（LAN）191およびワイドエリアネットワーク（WAN）193が含まれるが、その他のネットワークを含んでもよい。このようなネットワーク環境は、オフィス、企業規模のコンピュータネットワーク、イントラネット、およびインターネットで一般的に見られる。

【0027】LANネットワーク環境で使用する場

合、コンピュータ120は、ネットワークインタフェースすなわちアダプタ190を通じてLAN191に接続する。WANネットワーク環境で使用する場合、コンピュータ120は通例モデム192か、またはインターネットなどのWAN193を紹介して通信を確立するための他の手段を含む。モデム192は、内蔵型でも外付け式でもよく、ユーザ入力インタフェース180または他の適切な機器を紹介してシステムバス141に接続することができる。ネットワーク環境では、コンピュータ120との関連で図示するプログラムモジュール、またはその一部をリモートのメモリー記憶装置に記憶することができ。例として図4に、リモートアプリケーションプログラム195をリモートコンピュータ194に常駐プログラム195として示しているが、これに限定しない。図のネットワーク接続は例示的なものであり、コンピュータ間

に通信リンクを確立する他の手段を使用してよいことは理解されよう。

【0028】図5に、本発明で実施することのできるウェブサーバ202の構成を示す。一般に、ウェブサーバ202に記憶された情報には、モバイルデバイス30（本明細書では、入力の形態に基づき、適量、表示画面、マイクロフォン、カメラ、タッチセンシティブパネルなどを有する他形態のコンピュータ）に記憶された情報（図示せず）を通じて、または情報や音声により、またはキーを押すのに応答して電話機80が生成するトーンを通じて要求する電話機80を通じてアクセスすることができ。電話機80の場合には、ウェブサーバ202からの情報を音声によりユーザに提供する。

【0029】より重要なのは、情報をデバイス30を通じて得るか、または音声認識を用いて電話機80を通じて得るかに関係なく、単一の記録サーバ204がどちらの動作モードもサポートすることができる点でアーキテクチャ200が統一されていることである。さらに、アーキテクチャ200は、周知のマークアップ言語（例えばHTML、XML、eHTML、eXML、WMLなど）の拡張を使用して動作する。したがって、ウェブサーバ202に記憶された情報には、これらのマークアップ言語で利用される周知のGUI方式を用いてアクセスすることもできる。周知のマークアップ言語の拡張を使用することにより、ウェブサーバ202でのオーサリングが容易になり、現在存在するレガシーアプリケーションも、音声認識を含むように容易に修正することができる。

【0030】一般に、デバイス30は、ウェブサーバ202が提供するHTMLページ、スクリプトなどを実行する。一例として、音声（voice）認識が必要な場合には、デジタル化したオーディオ信号または音声特徴などの音声データ（オーディオ信号は上記のようにデバイス30で前処理する）を、音声認識中に使用する文法または言語モデルの指示とともに、記録サーバ204に提供する。記録サーバ204の実施態様は多くの形態をとることが可能であり、そのうちの1つを図示したが、一般にはコグナイズ211を含む。認識の結果は、所望の場合、または適切な場合にはローカルのレンダリングのためにデバイス30に送られる。認識と、使用する場合には任意のグラフィカルユーザインタフェースを通じて情報を提供すると、必要な場合には、デバイス30はその情報をウェブサーバ202に送信し、そこでさらに処理を行い、さらにHTMLページ/スクリプトを返す。

【0031】図5に示すように、デバイス30、ウェブサーバ202、および記録サーバ204は共通に（commonly）接続されており、また本明細書ではインターネットなどのワイドエリアネットワークであるネットワーク205を通じて個別にアドレス指定することができ。したがって、これらの装置はいずれも物理的に相互に接続して配置する必要はない。特に、ウェブサーバ202が記録サーバ204を含む必要はない。この方式による、ウェブサーバ202におけるオーサリングを、それが行うべきアプリケーションに集中させることができ、オーサ（author）は記録サーバ204の複雑性を知る必要がない。記録サーバ204は、独自に設計してネットワーク205に接続することができ、それによりウェブサーバ202でさらに変更を行わなくとも更新および改良することができる。下記で説明するよう、ウェブサーバ202は、クライアント側のマークアップおよびスクリプトを動的に生成することのできるオーサリング機構も含むことができる。別の実施形態で

は、英装マシンの能力に応じて、ウェブサーバ202、認識サーバ204、およびクライアント30を組み合わせる。図13

10032) 例えば、クライアントがパーソナルコンピュータなどの汎用コンピュータを含む場合には、クライアントは認識サーバ204を含むことができる。同様

に、所望の場合には、ウェブサーバ202および認識サ

ーバ204を単一マシンに組み込むことが可能である。

10033) クライアントデバイスに関して、クライ

ント/サーバシステムで入力データを処理する方法は、

クライアントデバイスのユーザから入力データを得るよ

うに構成された拡張を有するマークアップ言語ページを

サーバから受信すること、クライアントデバイスでマ

ークアップ言語ページを実行すること、入力データ

(ユーザから得た音声、DTMF、手書き、ジェスチ

ャ、または画像を表す) およびそれに関連する文法をク

ライアントからリモートに位置する認識サーバに送信す

ること、認識サーバからの認識結果をクライアントで

受信することを含む。クライアント/サーバシステム

のクライアントデバイスで実行するマークアップ言語を

有するコンピュータ読み取り可能な記憶媒体を提供す

ことができ、このマークアップ言語は、そのクライ

ントデバイスで入力される入力データと関連付けられた

指示する命令を有する。

10034) 電話機80を通じてウェブサーバ202へ

のアクセスは、配線式または無線式の電話網208へ

の電話機80の接続が含まれ、この電話網が電話機80

をサーバ204のゲートウェイ210に接続する。デ

ータウェイ210は、電話機80を電話音声ブラウザ2

12に接続する。電話音声ブラウザ212は、電話機80

に接続する。電話音声ブラウザ214は、音声ブ

ラウザ216を含む。デバイス30と同様に、電話音声

ブラウザ212は、ウェブサーバ202からHTMLペ

ージ/スクリプトなどを受信する。より重要なのは、こ

れらのHTMLページ/スクリプトが、デバイス30に

提供されるHTMLページ/スクリプトと同様の形態で

あることである。この方式によると、ウェブサーバ20

2は、デバイス30と電話機80を個別にサポートする

必要がなく、さらには標準的なGUIクライアントを個

別にサポートする必要もない。むしろ、共通のマークア

ップ言語を使用することができる。さらに、デバイス3

0と同様に、電話機80から送信可能な可読信号からの

音声認識は、ネットワーク205、または例えばTCP

/IPを使用する専用回線207を通じて、音声ブラウ

ザ216から認識サーバ204に提供される。ウェブサ

ーバ202、認識サーバ204、および電話音声ブラウ

ザ212は、図4に示す汎用デスクトップコンピュータ

など任意の適切なコンピュータ環境に実施するこ

とができる。

10035) ただしDTMF認識を用いる場合は、この

る。一般に、このような形態のマークアップ言語で一般

的なように、コードは本体部分260とスクリプト部分

262を含む。本体部分260は、実行するアクション

のタイプ、使用するフォーム、各種の情報フィールド2

50、252、および254を指定するコードの行を含

み、また提出ボタン264 (図6) 用のコードも含む。

この例は、イベントサポートと、埋め込まれたスクリ

プトスクリプトも表しており、提出ボタン264が起動

されると、スクリプト部分262で関数「verifi

y」が呼び出され、または実行される。「verifi

y」関数は、各クレジットカード (Visa, Mast

erCard, American Express) の

カード番号の長さが必要ながであることを確認する。

10040) 図8は、音声認識を使用してウェブサーバ

204に提供されるクレジットカード情報を得るための、

図6と同じGUIを生成するクライアントマークアップ

を表している。音声認識については下記で図8〜16と

の関連で説明するが、本明細書で説明する技術は、手書

を認識、ジェスチャ認識、および画像認識にも同様に

用できることを理解されたい。

10041) 一般に、エクステンション (拡張：一般に

は「タグ」としても知られる) はXML要素の小セット

であり、関連する属性およびDOMオブジェクトプロパ

ティ、イベント、およびメソッドを含み、ソースマ

ークアップドキュメントと合わせて使用することにより、認

識インタフェース、DTMFまたは呼称制御をソースベ

ージに適用する。エクステンションの形式 (formal

ity) および意味 (semantics) はソースド

キュメントの性質に依存しないので、エクステンション

は、HTML、XHTML、cHTML、XML、WM

Lで、あるいは任意の他のSGML由来のマークアップ

とともに等しく効果的に使用することができ、エク

ステンションは、階層的に使用することが可能な新しい機能オ

ブジェクトまたは要素を提供するドキュメントオブジェ

クトモデルに従う。各要素には属性、プロパティ、メン

ト、イベント、および/または他の「子」要素を含むこ

とができる。

10042) 本明細書で、エクステンションは、ブラウ

ザを実行するデバイスの機能に応じて、異なる2つの

「モード」で解釈できることに留意されたい。第1の

モード「オブジェクトモード」では、全機能を利用する

ことができる。アプリケーションによるエクステンショ

ンのプログラム上の操作は、そのデバイスのブラウザが

使用可能にするどの機構でも実行することができる。こ

れには、例えば、XHTMLブラウザにおける「Scr

ipt Interpreter」アプリケーションなどがある。この理由か

ら、エクステンションのオブジェクトおよびメソッド

の小セットだけを定義すればよく、これらは、デバイ

すなわちクライアント側に存在する任意のプログラム機

構によって操作される。オブジェクトモードは、イベ

ントおよびスクリプティングを提供し、またより多

くの機能を提供して、ダイアログのオーサに、音声対

話に対するより細かなクライアント側におけるコンロー

ルを与えることができる。本明細書で使用する場合は、フ

ラを「アプレベルブラウザ」と呼ぶ。この形のブラウ

ウザは、エクステンションのすべての属性、プロパ

ティ、メンツド、およびイベントをサポートする。アッ

プレブラウザは、通例、より高い処理能力を持つデバ

イスで使用する。

10043) エクステンションは、「宣言モード」でも

サポートすることができ、本明細書で使用する場合、

宣言モードで動作するブラウザを「ダウンレベルブラウ

ザ」と呼び、これは完全なイベントリングおよびスクリ

プティング機能はサポートしない。代わりにこの形のブ

ラウザは、所与のエクステンションの宣言的側面 (すな

わちコア要素および属性) をサポートするが、DOM

(ドキュメントオブジェクトモデル) オブジェクトのプ

ロパティ、メンツド、およびイベントのすべてはサポ

ートしない。このモードは専ら宣言構文だけを用い、さら

に、SMIL (同期化マルチメディア統合言語) 2.0

などの宣言マルチメディア同期化および協調機構 (同期

マークアップ言語) と併せて使用することができる。ダ

ウンレベルブラウザは、通例、処理能力が限られたデバ

イスで使用する。

10044) ここで、特定の入力モードについて論じて

おきたい。詳細には、音声認識を、少なくともディ

デバイスと併せて、そして別の実施形態ではポイント

デバイスとも併せて使用して、データ入力フィールドを

指定すると特に有用である。具体的には、このモードの

データ入力では、ユーザは一般に、いくつかのフィールド

を選択し、それに対応する情報を提供するのを制御すること

ができる。例えば、図6の例では、ユーザはまずフィー

ルド252にクレジットカード番号を入力し、次いでフ

ールド250にクレジットカードの種別を入力し、最

後にフィールド254に有効期限日を入力することがで

きる。同様に、ユーザは、所望の場合にはフィールド2

52に戻り、誤った入力訂正することもできる。下記

で説明するように音声認識と組み合わせると、平易で自

然なナビゲーション形態が提供される。本発明で使用する

場合、フィールドを自由な形で選択することを可能に

する画面表示と、音声認識の両方を使用するこの形の入

力を「マルチモーダル」と呼ぶ。

10045) 再び図8を参照すると、HTMLマークア

ップ言語のコードが示されている。図7に示すHTML

コードと同様に、このコードも、本体部分270および

スクリプト部分272を含んでいる。また図7に示すコ

ードと同様に、図8に示すコードは、フォームの位置お



明細欄に例示する実施形態では、音声認識が終了し、結果をクライアントに送り返すと、クライアントは `recognizeObject` を非同期化して、認識済みのテキストをそのオブジェクトと関連付ける。コード部分 2 2 8 2 に対しては 2 6 8 4 もこれと同様に動作し、フィードバック 5 2 5 および 2 5 4 ごとに固有の `recognizeObject` による呼び出しを受け取り、それをフィードバック 2 5 2 および 5 4 4 それぞれに関連付ける。カード番号フィードバック 2 5 2 の受信について、明したと同様で、カード番号フィードバックからカード番号の取得を認識する。

【0049】一般に、アーキテクチャ2000およびユーザインタラクションのマークアップ言語と併せた音声認識の使用は、次のように行われる。まず、音声と関連付けられたフィールドを指示する。図4の開始形態ではスタイルス333を使用するが、本発明はスタイルス333の使用例に限定するものではなく、ボタン、マウスポインタ、回転ホイールなど任意形態の指示を使用できることは理解されよう。周知のように、視覚的なマークアップ言語を使用して、「onClick」などそれに対応するイベントを提示することができる。本発明は、音声、手動、ジェスチャなどのコマンドの使用だけに限定しない。「onSelect」イベントの使用だけに限定しない。同じ目的に使用することができる。一実施形態では、このようなイベントリングは、それに対応する音声の開始および/または終わりの両方を示す制御を果たすので、特に有用である。また、音声の対象とするフィールドは、ユーザの対話を追跡するブラウザ上で実行されるブラウザによって、ユーザによって指定できることに留意されたい。

【0050】ここで注意したいのは、異なる音群認識システムにおいては、認識サーバ204の異なる振る舞いおよび/または出力が必要となることである。認識プロセスの開始はすべての場合に標準的なものであり、すなわちアップレブルラウザが明示的なstart()の呼び出しであり、あるいはダウンレベルラウザが重畳し出しており、あるいはダウンレベルラウザが重畳的callrec>要求であるが、音群認識を中止する手段は異なる可能性がある。

【0051】上記の例では、マルサモダールアプリケーションのユーザは、例えば圧力を感知するディスプレイを強く叩き、接触状態を保持することにより、デハイスへの入力制御する。するとプラウザは、例えば「pen-on」イベントのGUIイメントを使用して、認識をいつ中止するかを制御し、その後それに対応する結果を戻す。ただし、電話アプリケーション（下記で説明する）あるいは半使用せずに読むアプリケーションにといった苦みの決定はユーザー一切待たず、認識サーバー204またはク

ライアント30が、いつ認識を中止して結果を戻すか(通例は、文法中のパスを認識した時点)を決定する責任を負わなければならない。さらに、認識を中止する前に中間の結果を戻す必要があるディクテーションや他のシナリオ(「オープンマインドプロファイル」)としても知られている場合には、明示的な中止機能が必要とされるだけでなく、認識プロセスを中止する前に複数の認識結果をクライアント30および/またはウェブサーバ202に戻す必要がある。

【0052】一実施形態では、Reco要素は、下記の3つの認識モードを区別する「mode」属性を含むことができ、これにより認識カーバ204に、1つのように結果を戻すかを命令する。結果を戻すことは、「onReco」イベントを提供する、または「bind」要素を適宜起動することを意味する。一実施形態では、モードを指定しない場合、デフォルトの認識モードは「自動」にすることができ、

【0053】図14は、音響認識の「自動」モードの動作を模式的に表したものである（他の形態の認識にもこれと同様のモード、イベントなどを提供することができ、図8のスケジュール281は、認識サーバ204についての認識の開始283を指示するが、認識サーバ204がどこで音声を検出し（285）、その音声が続いたこと（287）を判定するを表している。

【0054】Reco 要素の各属性の属性性は、認識サーバ 2040 の属性性を制御する。属性性 [initial time out] 289 は、認識の開始 283 から音声の検出 285 までの間の時間である。この期間を超えることなく、[on silence] イベント 291 が認識サーバ 2040 から提供され、認識が中止されたことを知らせる。認識サーバ 2040 が、音声認識不可能であると判断した場合は、[on no recognition] イベント 293 を発行するが、これも認識を中止したことを示す。

[0055] 認識を中止またはキャンセルすることができ、他の属性には、「bubbleTimeoul」風性295があるが、これは285の音程の検出後に認識サーバ204で結果を返さなければならない期間である。この期間を超えると、エラー発生の影響に応じて異なるイベントが発行される。例えば、例外的に音声強度が急激に変化した場合、認識サーバ204がなおオーディオ処理

を行っている場合は、「onNoRecall」属性(2930)を発行する。しかし他の何らかの理由で「bubbleTimeout」属性(295)を超えた場合は、記録エラールタイムの可能性が高くなり、「onTimeout」イベント(297)が発行される。同様に「maxTimeout」属性(299)も提供することができ、これは、記録の開始(283)から結果をクライアント(30)に返すまでの期間(297)を超え、この期間を超えると、「onTimeout」イベント(297)が発行される。

【0056】ただし、「endSilence」属性(300)

0.1以上の期間を超えた場合、これは総裁が完了していることを示唆せず、この場合は総裁サーバー204が自動的に総裁を中止し、その結果を戻す。総裁サーバー204は、信頼度の測定を調整して、総裁結果を戻すべしと4うかがを判定できることに留意されたい。信頼度の測定値が閾値を下回る場合は、「onNoReco」属性2993を発行し、一方信頼度の測定値が閾値を上回る場合は、「onNoReco」属性303および総裁結果を発行する。したがって図4は、「自動モード」で、明示的なstop()の呼び出しが行われていない状況を表している。

【0057】図15は、認識サーバ204の「シングルスレッド」の動作を図式的に表したものである。「自動モード」の動作と関連で説明した属性およびイベントを適用することができ、したがって同じ参照番号で示している。しかし、この動作モードでは、stop（）呼び出し305を、スケジューラ281上に示している。「ペンstop（）呼び出し305は、ユーザによる「ペンアップ」などのイベントに相当する。この動作モードでは、認識結果を戻すことは、明示的なstop（）呼び出し305によって制御される。すべての動作モードの場合と同じ、「fontSilence」イベント291は、finalTimeOut期間289内に音声が発生されない場合に実行されるが、この動作モードでは認識を中止しない。同様に、stop（）呼び出し305以前の認識不可能な発声によって生成されるfontNoRec イベント293によっても認識は中止される。ただし、bubbleTimeOut イベント295または「maxTimeOut」属性299と関連付けられた期間を超えた場合は、認識を中止する。

【0058】図16は、認識サーバ204の「複数モード」の動作を模式的に表している。上記で指摘したように、この動作モードは、「オープンマイクロフォン」または、「ディクテーション」のシナリオで使用する。一般に、この動作モードでは、明示的なstop（）呼び出し305が受け取られるか、または「bubble timeout」性295または「max timeout」風性299に到達付けられた期間を越えるまで、間隔を置いて認識結果を戻す。ただし、「onSilence」イベント291、「onReco」イベント303、または「onNoReco」イベント299のいずれかが発生するまでは、この動作モードは継続する。

発生すること、これらによって(変換)エラーを回避(「max T (bubble time out) 期間が正しくなく「max T (imeout) 期間のタイマがリセットされることに留意されたい。

【0059】 一般に、この動作モードでは、stop ( ) 呼び出し305が受け取られるまで、記録されるフレーズごとに、「onRec イベント303を発行し、結果を戻す。記録不可能な音声のために「onS

リプトを含むHTMLとして実施する音声のみによるマークアップ言語を示す。図に明瞭に示すように、このコードも本体部分300およびサブ部分302を含む。このコードでは、マークアップ言語の別のエクステンション、すなわちパーズインなどの属性を含むプロンプトコントロール303がある。ただし、図9および10の音声のみの実施形態では、音声認識を別の方式で行う。この場合は、プロセス全体を、未入力(unfilled)および新しいオブジェクトを起動するスク립ト関数(c heckfilled)によって制御する。しかし、上記で図8との関連で説明したのと同じコンテキストを用いて文法を起動し、音声データおよび使用する文法の指示を認識サーバ204に提供する。同様に、認識サーバ204から受け取った出力を、クライアント(この場合は電話音声ブラウザ212)のフィールドと関連付ける。

【0064】一般に音声のみのアプリケーションに固有の他の機能は、音声認識されなかった際にユーザにそれを知らせることである。図8のようなマルチモーダルアプリケーションでは、fonNoRecは、表示されるフィールドに単にヌル値を入れて、認識が行われなかったことを示すので、それ以上の動作は必要とされない。音声のみの実施形態では、fonNoRec ol305は関数fnumbleを呼び出し、または実行する。この関数は、単語のフレーズを認識サーバ204に転送し、このフレーズは適切なテキストから音声に変換される。認識サーバ204は、オーディオストリームを電話音声ブラウザ212に戻し、次いでユーザが聴くためにそれを電話機80に送信する。同様に、音声のみのアプリケーションに実施するその他の波形プロンプトも、必要な場合には認識サーバ204によりオーディオストリームに変換する。

【0065】この例では、関数(welcome)を介してwelcomeプロンプトを再生すると、関数(heckfilled)がユーザに各フィールドを指示し、適切な文法を起動する。これには、入力されたフィールドを反復して、その情報が正しいことを確認することが含まれ、また(confirmation)文法の起動が含まれる。この実施形態では、各コントロールは、先の例の本体部分ではなくて、スク립ト部分302から開始されることに留意されたい。

【0066】マークアップ言語は、異なるタイプのクライアントデバイス(例えば、マルチモーダル、および電話機のような非表示式、音声入力ベースのクライアントデバイス)で実行することができ、各クライアントデバイスと対応するウェブサーバのために、音声に関連するイベント、GUIイベント、および電話イベントのうち少なくとも1つを統一する。これは、ウェブサーバ

【0065】この例では、関数(welcome)を介してwelcomeプロンプトを再生すると、関数(heckfilled)がユーザに各フィールドを指示し、適切な文法を起動する。これには、入力されたフィールドを反復して、その情報が正しいことを確認することが含まれ、また(confirmation)文法の起動が含まれる。この実施形態では、各コントロールは、先の例の本体部分ではなくて、スク립ト部分302から開始されることに留意されたい。

【0066】マークアップ言語は、異なるタイプのクライアントデバイス(例えば、マルチモーダル、および電話機のような非表示式、音声入力ベースのクライアントデバイス)で実行することができ、各クライアントデバイスと対応するウェブサーバのために、音声に関連するイベント、GUIイベント、および電話イベントのうち少なくとも1つを統一する。これは、ウェブサーバ

【0066】図9および10は、本発明でページ/スク

レイケーションのかなりの部分を、汎用的に、あるいはクライアントデバイスタイプのタイパに依存せずに書くことを可能にするので特に有用である。handle関数を含む一例を図8、および図9、10に示す。

【0067】図9、10には示していないが、このマークアップ言語には、電話機能をサポートするエクステンションがさらに2つある。すなわち、DTMF(デュアルトーン変調周波)制御、呼制御の要素またはオブジェクトである。DTMFは、recoコントロールと同様の働きをする。これは、キーパッドストリングからテキスト入力への単純な文法マッピングを指定する。例えば、「1」は食品品部門を意味し、「2」は薬品部門を意味するなどである。一方、呼オブジェクトは、呼の転送や第三者の呼出しのような電話機能を扱う。属性、プロパティ、メソッド、イベントについては付録で詳細に説明する。

【0068】図11および12は、音声のみの動作モードに適したマークアップ言語のさらに別の例を示す。この実施形態では、ユーザは、情報をいつ入力するか、または話すことに関する程度の制御権を有することができ、あるいは他の方法で発話を開始するようにユーザに指示することができるが、ユーザは当初要求されるよりも多くの情報を提供することができる。これは、「混合主導型」の一例である。一般に、この形のダイアログ対話では、ユーザはダイアログの主要なテーマと分かち合うことができる。上記で述べ、下記で詳細に説明する、ユーザがプロンプトに要求されるよりも多くの情報を提供する例のほかに、ユーザはその指示がないときにタスクを切り替えることもできる。

【0069】図11および12の例では、do\_filed)と識別する文法は、文法(g\_card\_unity pes)、(g\_card\_unity)、および(g\_expiry\_date)と関連付けられた情報を含む。この例では、電話音声ブラウザ212は、fonRec o)として示す認識済みの音声を受け取り、電話機80から受け取った音声データと、do\_filed)文法の使用の指示を認識サーバ204に送信し、関数(handle)が呼び出され、または実行されるが、これには音声データから認識されたフィールドの一部またはすべての値を関連付けることが含まれる。すなわち、認識サーバ204から得られる結果は、各フィールドに05で指定されるパイル規則に従って対応するフィールドと関連付ける。図5に示すように、認識サーバ204は、パラサ309を含むことができる。

【0070】図7、8、9、10、11、および12から、非常に類似したウェブ開発フレームワークを使用する。データの提示も、これらの各場合で非常に類似している。さらに、データ提示とフロー制御を分離すること





30

<bind...> 認識結果の処理  
 <dmf...> DTMFの構成および制御

10093] 2. Reco  
 Reco要素は、可能なユーザー入力と、入力結果の処理手段とを指定するのに使用する。したがって、その主要な要素は<grammar>および<bind>にすることができ、またレコグナイザプロパティを構成するためのリンクスを含む。

10094] Reco要素は、アププレベルブラウザではStarおよびStopのメソッドを介してプログラマ的に、またはSMILを使用できるブラウザではSMILコマンドを使用して起動する。この要素は、ダウンレベルブラウザ（すなわちスクリプトをサポートしないブラウザ）では、それがページ上にあることにより自動的にアクティブであると見なし、複数の文法を並行して起動することができ、複数のReco要素と同時にアクティブと見なすことができる。

10095] Recoは特定のモード、すなわち「自動」「シグナル」または「複数」をとることができ、これによりそれが使用可能にする認識シナリオの種類と、認識ブラッフフォームの振る舞いを区別する。

10096] 2. 1 Recoの内容

Reco要素は、1つまたは複数の文法と、任意選択で、認識結果を調べ、関連性のある部分をを含むページ中の面にコピーするバインド要素のセットを含む。

10097] アププレベルブラウザでは、Recoは、プログラマ的な起動、および個々の文法規則の非活動化をサポートする。指定しない場合は、ある認識コンテキストについて、文法のすべての最上位の規則がアクティブになることに留意されたい。

10098] 2. 1. 1 <grammar>要素

文法要素は、インラインの、またはsrc属性を使用し、参照する文法を指定するのに使用する。通例は少なくとも1つの文法（インラインまたは参照）を指定する。インライン文法はテキストベースの文法形式にすることができ、参照文法は、テキストベースまたはバイナリタイプにすることができ、複数の文法要素を指定することが可能である。複数の文法要素を指定する場合は、文法の規則を追加規則として同じ文法中に追加する。同じ名前の規則がある場合にはそれに上書きする。

10099] 属性：  
 ·src: インライン文法を指定する場合は任意選択。含める文法のURL。指定しない場合は、ある認識コンテキストについて、文法のすべての最上位規則がアクティブになることに留意されたい。

10100] ·langID: 任意選択。音声エンジンが使用する言語を示すストリング。ストリングの形式は、xml:lang定義に従う。例えば、lang

37

ID="en-us"は、米国英語を要す。この属性は、langIDを文法URL中で指定しないときにのみ有効である。指定しない場合は、米国英語を使用する。

10101] langIDが複数の箇所で指定される場合、langIDは、最低の有効範囲からの優先順位に従う。すなわち、リモートの文法ファイル（つまりその文法ファイル中で指定される言語ID）、次いで文法要素、次いでreco要素の順となる。

(grammar src="Prodcty.rml" /)

または

(grammar  
 (rule lptel="active")  
 (p)から(p)  
 (ruleref name="cities" /)  
 /rule)  
 (rule name="cities")  
 /rule)

(1)  
 (p) ケンブリッジ (p)  
 (p) シアトル (p)  
 (p) ロンドン (p)  
 /1)  
 /rule)  
 /grammar)

20

srcで参照する文法とインライン文法の両方を指定\*

(xml confidence="40"  
 <travel text="シアトルからボストンまで行きたい">  
 <origin\_city  
 <dest\_city confidence="45"> シアトル  
 </dest\_city>  
 </travel>  
 </xml>

40

文法中（in-grammar）認識は、セマンティックマークアップ言語すなわちSMILでXMLドキュメントを生成することになっているので、SMILドキュメントからバインドする値は、XPathクエリを使用して参照する。また、値をバインドするページ中の要素（これはフォーコントロールである可能性が高い）は一意に識別すべきなので、これらのターゲット要素は直接参照する。

10104] 属性：  
 ·targetElement: 必須。SMILからvaueの内容を割り当てて要素（W3C SMIL2.0と同様）。  
 10105] ·targetAttribute: 任意選択。SMILからvaueの内容を割り当ててターゲット要素の属性（SMIL2.0のattribute Name属性と同様）。指定しない場合は、「value」になる。

(18) 特開2003-67177

33

```

(Input name="iriBotOrigin" type="text"/>)
(Input name="iriBotDest" type="text"/>)

(rec id="travel")
(grammar src="/city.xml" /)

(bind target|elem="iriBotOrigin"
value="/orig|city" /)
(bind target|elem="iriBotDest"
value="/dest|city" /)

(/reco)

```

このイベントは、バインド操作の事前条件として、des-  
cityの結果の信頼度属性にテストを行う以下列の例  
のように条件付きの場合もある。

```

(bind target|elem="iriBotDest"
value="/dest|city"
test="/snl|dest|city[confidence >= 40]" /)

```

バインド要素は、ダウンレベルまたはアップレベルのブ  
ラウザで認識結果を処理する単純な宣言的手段である。

20 より複雑な処理の場合、アップレベルブラウザによって  
サポートされるrecoDOMオブジェクトは、onRe-  
coイベントハンドラを実装して、プログラマ的なス  
クリプト分析と認識の戻しの後処理を行えるようにす  
る。

[0109] 2. 2 属性およびプロパティ  
以下の属性はすべてのブラウザでサポートされ、プロパ  
ティはアップレベルブラウザによってサポートされる。

[0110] 2. 2. 1 属性

以下のRecoの属性は、ダイアログターンのために音  
声レコグナイザを構成するのに使用する。

[0111]・initialTimeout: 任意選  
択。認識の開始から音声の検出までのミリ秒単位の時  
間。この値は認識プラットフォームに渡され、これを超  
えた場合は、onSilenceイベントが認識ブラッ  
クフォームから提供される(2. 4. 2参照)。指定し  
ない場合、音声プラットフォームはデフォルト値を使用  
する。

[0112]・bubbleTimeout: 任意選  
択。音声の検出後にレコグナイザが結果を返さなければ  
ならないミリ秒単位の期間。自動モードおよびシングル  
モードのrecoの場合、これは音声検出からstop  
呼び出しまでの期間に該当する。「指数」モードのre-  
coの場合、このタイムアウトは、「指数」モードの各認識  
の戻しまでの期間に相当する。すなわち、各結果の戻し  
または他のイベントの後にこの期間を再び開始する。こ  
のタイムアウトを越えると、エラーの発生の有無に応じ  
て異なるイベントを投入する。例えば、音声が無効的に  
長い場合など、レコグナイザがおオーディオを処理し  
ている場合は、ステータスコード13により(2. 4.

(19) 特開2003-67177

34

33

[0118] 2. 2. 2 プロパティ  
以下のプロパティは、認識プロセスによって戻される結  
果を含む(これはアップレベルブラウザにサポートさ  
れる)。

[0119]・recoResult: 読み取り専用。  
認識の結果、2. 1. 2で述べたように、セマンティッ  
クマークアップ言語(SML)を含むXML DOMノ  
ードオブジェクト中に保持される。認識が行われなかつ  
た場合、このプロパティはヌルに戻る。

[0120]・text: 読み取り/書き込み。認識さ  
れた単語のテキストを保持するストリング(すなわち、  
読み取りモードにおけるrecoResult中のSM  
L認識の戻しの中の最上位要素のテキスト属性の内容を  
取り省略表現)。書き込みモードでは、ストリングを割  
り当てることができ、次いでそのストリングが認識結果  
に対応するものとしてそれを構文解析する。書き込みモ  
ードでは、このマークアップタグおよびその処理を、ク  
ライアントデバイスの他のコンポーネントまたはアプリ  
ケーションに拡張することができる。このストリン  
グは、「smex」メタセージオブジェクトから得られ  
る。

[0121]・status: 読み取り専用。認識ブラ  
ックフォームが返すステータスコード。可能な値は、認  
識が成功した場合の0、あるいは障害値-1から-4  
(Startメソッド(節2. 3. 1)およびActi-  
vateメソッド(節2. 3. 4)で可能な例外で定義  
する)、およびレコグナイザイベントを受け取った際に  
セットされるステータス-11から-15(2. 4参  
照)。

[0122] 2. 3 オブジェクトメソッド  
recoの起動および文法の起動は、RecoのDOM  
オブジェクト中の以下のメソッドを使用して制御するこ  
とができる。これらのメソッドにより、アップレベルブ  
ラウザはRecoオブジェクトの開始および中止、進行  
中の認識のキャンセル、個々の文法のトップレベルの規  
則の起動および非活性化を行うことができる(アップレ  
ベルブラウザのみ)。

[0123] 2. 3. 1 Start  
Startメソッドは、明示的に非活性化していない  
認識コンテキストについてのすべての最上位規則をアク  
ティブな文法として使用して認識プロセスを開始する。  
構文: Object. Start ( )  
戻り値: なし  
例外: このメソッドは、非ゼロのステータスコードをセ  
ットし、障害があった際はonNoRecoイベントを  
発生させる。可能性のある障害には、文法が存在しない  
(recoステータス=-1)、文法のコンパイルの失  
敗、存在しないURIなど様々な原因になりうる文法の  
ロードの失敗(recoステータス=-2)、あるいは  
音声プラットフォームのエラー(recoステータス=

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

-3

-4

-5

-6

-7

-8

-9

-10

-11

-12

-13

-14

-15

-16

-17

-18

-19

-20

-21

-22

-23

-24

-25

-26

-27

-28

-29

-30

-31

-32

-33

-34

-35

-36

-37

-38

-39

-40

-41

-42

-43

-44

-45

-46

-47

-48

-49

-50

-51

-52

-53

-54

-55

-56

-57

-58

-59

-60

-61

-62

-63

-64

-65

-66

-67

-68

-69

-70

-71

-72

-73

-74

-75

-76

-77

-78

-79

-80

-81

-82

-83

-84

-85

-86

-87

-88

-89

-90

-91

-92

-93

-94

-95

-96

-97

-98

-99

-100

-101

-102

-103

-104

-105

-106

-107

-108

-109

-110

-111

-112

-113

-114

-115

-116

-117

-118

-119

-120

-121

-122

-123

-124

-125

-126

-127

-128

-129

-130

-131

-132

-133

-134

-135

-136

-137

-138

-139

-140

-141

-142

-143

-144

-145

-146

-147

-148

-149

-150

-151

-152

-153

-154

-155

-156

-157

-158

-159

-160

-161

-162

-163

-164

-165

-166

-167

-168

-169

-170

-171

-172

-173

-174

-175

-176

-177

-178

-179

-180

-181

-182

-183

-184

-185

-186

-187

-188

-189

-190

-191

-192

-193

-194

-195

-196

-197

-198

-199

-200

-201

-202

-203

-204

-205

-206

-207

-208

-209

-210

-211

-212

-213

-214

-215

-216

-217

-218

-219

-220

-221

-222

-223

-224

-225

-226

-227

-228

-229

-230

-231

-232

-233

-234

-235

-236

-237

-238

-239

-240

-241

-242

-243

-244

-245

-246

-247

-248

-249

-250

-251

-252

-253

-254

-255

-256

-257

-258

-259

-260

-261

-262

-263

-264

-265

-266

-267

-268

-269

-270

-271

-272

-273

-274

-275

-276

-277

-278

-279

-280

-281

-282

-283

-284

-285

-286

-287

-288

-289

-290

-291

-292

-293

-294

-295

-296

-297

-298

-299

-300

-301

-302

-303

-304

-305

-306

-307

-308

-309

-310

-311

-312

-313

-314

-315

-316

-317

-318

-319

-320

-321

-322

-323

-324

-325

-326

-327

-328

-329

-330

-331

-332

-333

-334

-335

-336

-337

-338

-339

-340

-341

-342

-343

-344

-345

-346

-347

-348

-349

-350

-351

-352

-353

-354

-355

-356

-357

-358

-359

-360

-361

-362

-363

-364

-365

-366

-367

-368

-369

-370

-371

-372

-373

-374

-375

-376

-377

-378

-379

-380

-381

-382

-383

-384

-385

-386

-387

-388

-389

-390

-391

-392

-393

-394

-395

-396

-397

-398

-399

-400

-401

-402

-403

-404

-405

-406

-407

-408

-409

-410

-411

-412

-413

-414

-415

-416

-417

-418

-419

-420

-421

-422

-423

-424

-425

-426

-427

-428

-429

-430

-431

-432

-433

-434

-435

-436

-437

-438

-439

-440

-441

-442

-443

-444

-445

-446

-447

-448

-449

-450

-451

-452

-453

-454

-455

-456

-457

-458

-459

-460

-461

-462

-463

-464

-465

-466

-467

-468

-469

-470

-471

-472

-473

-474

-475

-476

-477

-478

-479

-480

-481

-482

-483

-484

-485

-486

-487

-488

-489

-490

-491

-492

-493

-494

-495

-496

-497

-498

-499

-500

-501

-502

-503

-504

-505

-506

-507

-508

-509

-510

-511

-512

-513

-514

-515

-516

-517

-518

-519

-520

-521

-522

-523

-524

-525

-526

-527

-528

-529

-530

-531

-532

-533

-534

-535

-536

-537

-538

-539

-540

-541

-542

-543

-544

-545

-546

-547

-548

-549

-550

-551

-552

-553

-554

-555

-556

-557

-558

-559

-560

-561

-562

-563

-564

-565

-566

-567

-568

-569

-570

-571

-572

-573

-574

-575

-576

-577

-578

-579

-580

-581

-582

-583

-584

-585

-586

-587

-588

-589

-590

-591

-592

-593

-594

-595

-596

-597

-598

-599

-600

-601

-602

-603

-604

-605

-606

-607

-608

-609

-610

-611

-612

-613

-614

-615

-616

-617

-618

-619

-620

-621

-622

-623

-624

-625

-626

-627

-628

-629

-630

-631

-632

-633

-634

-635

-636

-637

-638

-639

-640

-641

-642

-643

-644

-645

-646

-647

-648

-649

-650

-651

-652

-653

-654

-655

-656

-657

-658

-659

-660

-661

-662

-663

-664

-665

-666

-667

-668

-669

-670

-671

-672

-673

-674

-675

-676

-677

-678

-679

-680

-681

-682

-683

-684

-685

-686

-687

-688

-689

-690

-691

-692

-693

-694

-695

-696

-697

-698

-699

-700

-701

-702

-703

-704

-705

-706

-707

-708

-709

-710

-711

-712

-713

-714

-715

-716

-717

-718

-719

-720

-721

-722

-723

-724

-725

-726

-727

-728

-729

-730

-731

-732

-733

-734

-735

-736

-737

-738

-739

-740

-741

-742

-743

-744

-745

-746

-747

-748

-749

-750

-751

-752

-753

-754

-755

-756

-757

-758

-759

-760

-761

-762

-763

-764

-765

-766

-767

-768

-769

-770

-771

-772

-773

-774

-775

-776

-777

-778

-779

-780

-781

-782

-783

-784

-785

-786

-787

-788

-789

-790

-791

-792

-793

-794

-795

-796

-797

-798

-799

-800

-801

-802

-803

-804

-805

-806

-807

-808

-809

-810

-811

-812

-813

-814

-815

-816

-817

-818

-819

-820

-821

-822

-823

-824

-825

-826

-827

-828

-829

-830

-831

-832

-833

-834

-835

-836

-837

-838

-839

-840

-841

-842

-843

-844

-845

-846

-847

-848

-849

-850

-851

-852

-853

-854

-855

-856

-857

-858

-859

-860

-861

-862

-863

-864

-865

-866

-867

-868

-869

-870

-871

-872

-873

-874

-875

-876

-877

-878

-879

-880

-881

-882

-883

-884

-885

-886

-887

-888

-889

-890

-891

-892

-893

-894

-895

-896

-897

-898

-899

-900

-901

-902

-903

-904

-905

-906

-907

-908

-909

-910

-911

-912

-913

-914

-915

-916

-917

-918

-919

-920

-921

-922

-923

-924

-925

-926

-927

-928

-929

-930

-931

-932

-933

-934

-935

-936

-937

-938

-939

-940

-941

-942

-943

-944

-945

-946

-947

-948

-949

-950

-951

-952

-953

-954

-955

-956

-957

-958

-959

-960

-961

-962

-963

-964

-965

-966

-967

-968

-969

-970

-971

-972

-973

-974

-975

-976

-977

-978

-979

-980

-981

-982

-983

-984

-985

-986

-987

-988

-989

-990

-991

-992

-993

-994

-995

-996

-997

-998

-999

-1000

例外：なし  
[0128] 2. 4 Recoイベント  
Reco DOMオブジェクトは以下のイベントをサポートし、そのハンドラは reco要素の属性として指定することができる。  
[0129] 2. 4. 1 onReco: このイベントは、レコグナイザが、そのプラウザで利用することのできる認識結果を得ると起動される。自動モードの reco \*

例文：  
[0130]  
[表1]

インラインHTML イベントプロパティ デフォルトアクション	<Reco onReco="handler"> Object.onReco = handler; Object.onReco = GetRef("handler");
--------------------------------------	--

[0131] イベントオブジェクト情報： ※ [表2]  
[0132] ※

バブル 起動するには デフォルトアクション	なし ユーザが何かを扱う 認識結果オブジェクトを返す
-----------------------------	----------------------------------

[0133] イベントプロパティ：イベントハンドラはプロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる（下記の例のイベントオブジェクトの使用を参照のこと）。  
[0134] 例  
次のXMLの断片では onReco を使用して、認識結果を構文解析し、その値を適切なフィールドに割り当てるスクリプトを呼び出している。

```
(script) { [CDATA[  
function processClyRecognition () {  
    smlResult =  
event1.srcElement.recogResult;  
  
    origNode =  
smlResult.selectSingleNode("//orig_node");  
if (origNode != null)  
tx (origNode.value = origNode.text;  
  
    destNode =  
smlResult.selectSingleNode("//dest_node");  
if (destNode != null) tx (destNode.value  
= destNode.text;  
)  
}] } /script)
```

[0135] 2. 4. 2 onSilence: onSilenceは、RecoのinitialTimeout属性で指定された時間が過ぎる前に、認識プラットフォームが検出した無音声のイベントに対処する（2. 2. 1参照）。このイベントは、自動認識モードの認識プロセスを自動的にキャンセルする。  
例文：  
[0136]

50 [0136]

[表3]

インラインHTML イベントプロパティ (ECMAScript) デフォルトアクション	<Reco onSilence="handler"> Object.onSilence = handler; Object.onSilence = GetRef("handler");
--	---

[0137] イベントオブジェクト情報： \* [表4]  
[0138] \*

バブル 起動するには デフォルトアクション	なし initialTimeout 属性で指定される期間中にレコグナイザが 音声を検出しなかった ステータスを-1にセット
-----------------------------	--

[0139] イベントプロパティ：イベントハンドラは、プロパティを直接受け取ることはないが、ハンドラはデータについてイベントオブジェクトに照会を行うことができる。  
[0140] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。

[0141] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0142] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0143] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0144] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。

[0145] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0146] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0147] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0148] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。  
[0149] 2. 4. 3 onTimeout: 20 イベントは、認識プロセスを自動的にキャンセルする。

インラインHTML イベントプロパティ (ECMAScript) デフォルトアクション	<Reco onTimeout="handler"> Object.onTimeout = handler; Object.onTimeout = GetRef("handler");
--	---

[0144] イベントオブジェクト情報： \* [表6]  
[0145] \*

バブル 起動するには デフォルトアクション	なし 認識の中止時に、maxTime 属性で指定された期間が過ぎ るとアラウクが投入する neo ステータスを-12にセットする
-----------------------------	---

[0146] イベントプロパティ：イベントハンドラは、プロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。  
[0147] 2. 4. 4 onNoReco: onNoRecoは、有効な認識結果を返すことができない際に、音声認識プラットフォームが投入するイベント用のハンドラ

[0148] 2. 4. 4 onNoReco: onNoRecoは、有効な認識結果を返すことができない際に、音声認識プラットフォームが投入するイベント用のハンドラ  
[0149] 2. 4. 4 onNoReco: onNoRecoは、有効な認識結果を返すことができない際に、音声認識プラットフォームが投入するイベント用のハンドラ  
[0150] 2. 4. 4 onNoReco: onNoRecoは、有効な認識結果を返すことができない際に、音声認識プラットフォームが投入するイベント用のハンドラ

インラインHTML イベントプロパティ デフォルトアクション	<Reco onNoReco="handler"> Object.onNoReco = handler; Object.onNoReco = GetRef("handler");
--------------------------------------	--

[0149] イベントオブジェクト情報： [表8]  
[0150]

41	なし	なし
42	レコグナイザが音を検出するが、声を検察することができない	レコグナイザが音を検出するが、声を検察することができない
43	ステータス-13: 音が検出されたが、解釈できる音が入らなかった場合	ステータス-13: 音が検出されたが、解釈できる音が入らなかった場合
44	ステータス-14: いくらかの音が入ったが、解釈された音の開始については221の開始の属性を参照のこと	ステータス-14: いくらかの音が入ったが、解釈された音の開始については221の開始の属性を参照のこと
45	ステータス-15: 音が検出され、解釈されたが、音声の検出がbottleInaudible属性で指定された期間まで完全な検出を戻すことができなかった場合 221 参照	ステータス-15: 音が検出され、解釈されたが、音声の検出がbottleInaudible属性で指定された期間まで完全な検出を戻すことができなかった場合 221 参照

43  
element の必要な属性を指定することができる。これは、例えば、HTML フォームコントロール中の値をデリファレンスするのに有用である。下の例では、textInputBoxOrigin 要素およびtextInputBoxDestination 要素のvalue 属性を、プロンプトの出力前にテキストに挿入している。  
(prompt id="Confirm")  
あなたが行きたいのは  
(value targetElement="textInputBoxOrigin"  
targetAttribute="value" />  
から  
(value targetElement="textInputBoxDestination"  
targetAttribute="value" />  
ですか?  
(prompt)

44  
\* この値要素は、合成したプロンプトの代わりに、あるいはその中で再生するあらかじめ記録したオーディオファイルを参照するにも使用することができる。次の例では、プロンプトの最後にピープ音を鳴らしている。

[0159] 3. 1. 3 オーディオファイル \*  
(prompt)  
ピープという音がしたらメッセージを録音してください  
(value href="/wav/beep.wav" />  
(prompt)

[0160] 3. 1. 4 参照プロンプト  
インラインの内容を指定する代わりに、src 属性を型要素とともに使用し、URI を介して外部の内容を参照することができる。例えば、  
(prompt id="Welcome"  
src="/ACMEWelcomePromptIsWelcome" />

src 属性の対象は、インラインプロンプトに指定する上記の内容の任意部分またはすべてを保持することができる。  
[0161] 3. 2 属性およびプロパティ  
このプロンプト要素は、以下の属性 (ダウンレベルブラウザ) およびプロパティ (ダウンレベルおよびアップレベルブラウザ) を保持する。

[0162] 3. 2. 1 属性  
・bargein: 任意選択。テキストから音声への合成用のマークアップ言語タイプ。デフォルトは「SAP」。  
・src: インラインプロンプトを指定する場合は任意選択。参照するプロンプトのURI (3. 1. 4 参照)。

・bargain: 任意選択。整数。プロンプトの開始から、人間の聴者が再生を中断できるようにするまでのミリ秒単位の時間。デフォルトは無限、すなわちバージンを経可しない。bargain=0 にすると、即時のバージンが可能になる。これは、プラットフォームがサポートするどの種のバージンにも該当する。reco を開始する時間にとちを可能にするかに応じて、キーワードまたはエンゲージバースのバージン時間をこの方式で構成することができる。  
・preletich: 任意選択。ページをロードする際

30 から戻されるステータスコード。  
・innerlexi: 読み取り専用。このプロパティはプロンプトのテキストの複写 (transcript ion) を提供し、それがシンセサイザに送られる。例えば、あるプロンプトがオーディオウェーブファイルの再生を含む場合、このプロパティはそのプロンプトのテキストバージョン (オーディオウェーブファイルとともに記録することが多い) を提供し、これはその後、例えばクライアントデバイスで実行するコンポーネントまたはアプリケーションにプロンプトのテキストバージョンを提供することにより、表示するが、またはその他の形で使用することができる。また innerlexi プロパティを使用して、動的コンテンツを含むプロンプトのテキストバージョンも提供することができる。  
[0164] 3. 3 プロンプトメソッド  
プロンプトの再生は、プロンプトのDOMオブジェクト中の以下のメソッドを使用して制御することができる。この方式により、アップレベルブラウザは、プロンプトオブジェクトを開始および停止し、進行中のプロンプトを一時停止および再開し、合成音声のスピードおよび音量を要することができる。

41	なし	なし
42	レコグナイザが音を検出するが、声を検察することができない	レコグナイザが音を検出するが、声を検察することができない
43	ステータス-13: 音が検出されたが、解釈できる音が入らなかった場合	ステータス-13: 音が検出されたが、解釈できる音が入らなかった場合
44	ステータス-14: いくらかの音が入ったが、解釈された音の開始については221の開始の属性を参照のこと	ステータス-14: いくらかの音が入ったが、解釈された音の開始については221の開始の属性を参照のこと
45	ステータス-15: 音が検出され、解釈されたが、音声の検出がbottleInaudible属性で指定された期間まで完全な検出を戻すことができなかった場合 221 参照	ステータス-15: 音が検出され、解釈されたが、音声の検出がbottleInaudible属性で指定された期間まで完全な検出を戻すことができなかった場合 221 参照

[0151] イベントプロパティ: イベントハンドラはプロパティを直接受け取ることはないが、データについてこのイベントオブジェクトに照会を行うことができる。  
20  
プロンプト要素は、ダウンレベルブラウザによって宣言的に解釈する (あるいはSMILコマンドで起動する) ことも、アップレベルブラウザのオブジェクトメソッドによって宣言的に解釈することもできる。

[0153] 3. 1 プロンプト内容  
プロンプト要素は、テキストまたはオーディオファイルへのリファレンスの形で、あるいはこの両方の形でシテム出力用のリソースを含む。  
[0154] 簡単なプロンプトは、出力に必要なテキストだけを指定すればよい。例えば、  
(prompt id="Welcome")  
ACME 天気予報へのお電話ありがとうございます

※ クリックアップ言語でも使用することができる。 (この形式は、3. 2. 1 で説明する「lis」属性で指定することができる。) 次の例は、その中の特定の単語を強調する命令を含むテキストを示している。  
(prompt id="giveBalance")  
あなたの口座の残高は<emph>5ドル</emph>です

[0156] 3. 1. 2 動的な内容  
このプロンプトの実際の内容は、プロンプトの出力の直前にクライアントで計算する必要がある場合がある。例えば特定の値を決定するには、ある変数にその値をデリファレンスする必要がある。この値要素はこの目的に使用することができる。  
[0157] 値要素  
value: 任意選択。ドキュメント中の要素の値を返す。  
targetElement: 任意選択。href またはtargetElement を指定しなければなら

ない。取り出す値を含む要素のID。  
targetAttribute: 任意選択。値を取り出す要素の属性。  
href: 任意選択。オーディオセグメントのURI。両方ある場合には、href がtargetElement を上書きする。  
[0158] targetElement 属性は、それを含むドキュメント中の要素を参照するのに使用される。targetElement によってIDが指定された要素の内容を、合成するテキストに挿入する。所望の内容がその要素の属性に保持されている場合、targetAttribute を使用して、targetElement

40  
このプロンプトの再生は、プロンプトのDOMオブジェクト中の以下のメソッドを使用して制御することができる。この方式により、アップレベルブラウザは、プロンプトオブジェクトを開始および停止し、進行中のプロンプトを一時停止および再開し、合成音声のスピードおよび音量を要することができる。

再生がまだ中止されていない場合に再生を中止し、オーディオバッファをフラッシュする。再生がすでに中止されている場合、このメソッドは単にオーディオバッファをフラッシュする。

構文: `Object. Stop ( ) ;`

戻り値: なし

例外: なし

[0170] 3. 3. 5 Change

再生の速度および/または音量を変更する。Changeは再生中に呼び出すことができる。

構文: `Object. Change (speed, volume) ;`

パラメータ:

- ・ speed: 必須。変化させる係数。speed=2.0は、現在の速度を2倍にすることを意味し、speed=0.5は、現在の速度の2分の1にすることを意味し、speed=0は、デフォルト値に戻すことを意味する。
- ・ volume: 必須。変化させる係数。volume=2.0は、現在の音量を倍にすることを意味し、volume=0.5は、現在の音量を半分にすることを意味し、volume=0は、デフォルト値に戻すことを意味する。

戻り値: なし

例外: なし

[0171] 3. 3. 6 プロンプトコントロールの例

次の例は、キーワードバーゼインの機構をサポートしないブラウザに對して、上記のメソッドを使用するプロンプトコントロールをオーサリングする仕組みを示している。

[0172]

```

<html>
<title>プロンプトコントロール</title>
<head>
<script>
<!--
function checkKBBargein() {
  news.change (1.0, 0.5); // 騒音中は音量を下げる
  if (keyword.text == "") { // 結果が関値以下である場合
    news.change (1.0, 2.0); // 音量を元に戻す
  }
  keyword.start (0); // 騒音を再開
} else {
  news.stop (0); // キーワード検出！プロンプトを中止
  // 必要事項を行う
}
}
//
</script>
<script for="window" event="onload">
<!--

```

(25) news.start (0); // keyword.start (0);

```

//
</script>
</head>
<body>
<prompt id="news" bargein="0">
水曜日の株式市場も、投資家が、来週の連邦準備理事会の会合に先立ち大きな動きにつながる材料を得られなかったことから展開に活気がありませんでした。ハイテク銘柄中心のナスダック総合指数は42.51ポイント下落し、2156.26で取引を終えました。ダウジョーズ工業平均株価は、午後に入って反騰がなく17.05ポイント下落して10866.46で取引を終えました。

```

\*プロンプトDOMオブジェクトは以下のイベントをサポートするが、そのハンドラはプロンプト要素の属性として指定することができる。

[0174] 3. 4. 1 onbookmark

合成ブックマークに遭遇すると発生する。このイベントは再生を一時停止しない。

構文:

[0175]

[表9]

```

<!--
</prompt>

```

```

<reco id="keyword">
  reject="70"
  onreco="checkKBBargein()"
  </reco>
  </body>
  </html>

```

[0173] 3. 4 プロンプトイベント

イベントHTML	イベントプロパティ
インラインHTML	<prompt onbookmark="handler">
オブジェクトプロパティ	Object.onbookmark="handler"
オブジェクトプロパティ	Object.onbookmark=""
オブジェクトプロパティ	Object.onbookmark=""

[0176] イベントオブジェクト情報:

[0177]

バブル	なし
起動するには	レンダリングしたストリング中のブックマークに遭遇する
デフォルトアクション	ブックマークストリングを戻す

[0178] イベントプロパティ: イベントハンドラは、イベントを構成するかの決定はプラットフォームによるプロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。

[0179] 3. 4. 2 onBargein: ユーザのバーゼインイベントを検出すると発生する。(例えば、エネルギー検出やキーワード認識など、何がバーゼイン

インラインHTML	<prompt onBargein="handler">
イベントプロパティ	Object.onBargein="handler"
イベントプロパティ	Object.onBargein=""
イベントプロパティ	Object.onBargein=""



【0181】 イベントオブジェクト情報:

【0182】 \*

バブル	なし
発動するには	バージインイベントに通過する
デフォルトアクション	なし

【0183】 イベントプロパティ: イベントハンドラは ※シンプの再生が最後に達するか、または例外 (上記に定プロパティを直接受け取ることはないが、ハンドラはデ

ータについてこのイベントオブジェクトに照会を行うこ

とができる。

【0184】 3. 4. 3 onComplete: プロ ※ 【表13】

インラインHTML	Sound.onComplete="Handler">
イベントプロパティ	Object.onComplete="handler"
	Object.onComplete =
	GetRef("handler")

【0186】 イベントオブジェクト情報:

【0187】 \*

バブル	なし
発動するには	プロンプト再生が完了する
デフォルトアクション	再生が正常に完了した場合はステータス=0にセットし、 その他の場合は上記に指定するようにステータスをセッ トする

【0188】 イベントプロパティ: イベントハンドラは 正か目的地の提供のいずれかであるユーザ応答の意味を  
プロパティを直接受け取ることはないが、ハンドラはデ 判定する仕組みを示している。onBargeInハン  
ータについてこのイベントオブジェクトに照会を行うこ ドラが、プロンプト中に通過した最後のブックマークに  
とができる。 グローバルな「mark」変数を設定するスク립トを

【0189】 3. 4. 4 ブックマークおよびイベント 呼び出し、この「mark」の値をrecoの後処理関  
の使用 数 (「heard」) で使用して、正しい値をセットし

次の例は、プロンプトの出力中にバージインが行われた

場合に、ブックマークイベントを使用して、出発地の訂

<script><![CDATA[

var mark;

function interrupt() {

mark = event.srcElement.bookmark;

}

function ProcessCityConfirm() {

confirm.stop(); // オーディオバッファをフラッシュする

if (mark == "mark\_origin\_city")

txtBoxOrigin.value =

event.srcElement.text;

else

txtBoxDest.value =

event.srcElement.text;

}

]></script>

<body>

<input name="txtBoxOrigin" value="Seattle"

type="text"/>

<input name="txtBoxDest" type="text" />

<prompt id="confirm" onBargeIn="interrupt0"

bargeIn="0">

<bookmark mark="mark\_origin\_city" />

<value targetElement="origin"

targetAttribute="value" />から

<bookmark mark="mark\_dest\_city"

>行きたい行先地を書いて下さい

</prompt>

<reco onReco="ProcessCityConfirm0">

<grammar src="/gm/1033/cities.xml" />

</reco>

</body>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

</html>

city\_choice)を起動して、ユーザが1を

<input type="text" name="credit\_card\_number" />

<prompt onBookmark="dtmf.Start0; speech.Start0"

この例は、いかにしてユーザが複数フィールドに入力す  
るのを可能にするかを示している。

【0195】 例3: 音声入力およびDTMF入力をとも  
に許可し、ユーザがDTMFを開始した際に音声を使用  
不可にするには

```
33 bargein="0">
  <bookmark name="starting" />とか、またはあなたのクレジット
  カード番号を入力してください
  </prompt>
  <DTMF id="dtmf" escape="#" length="16"
interdigitTimeout="2000"
onkeypress="speech.Stop(0)">
  <bind targetElement="credit_card_number" />
</DTMF>
<recog id="speech">
  <grammar src="/grm/1033/digits.xml" />
  <bind targetElement="credit_card_number" />
</recog>
```

[0196] 4. 2 属性およびプロパティ  
読み取り専用ストリング。空白で分離されたトークンストリングを記述し、各トークンはDTMF文法に従って変換する。

4. 2. 1 属性  
・dtm(grammar: 必須、DTMF文法のURI。

[0197] 4. 2. 2 プロパティ  
・DTMF(grammar: 読み取りおよび書き込み、ストリング変換行列に対するDTMFを表すXML D 20 ない場合は、電話プラットフォームの内部設定になる。OM ノードオブジェクト (DTMF文法とも呼ぶ)。デフォルト文法は、

```
(dtlgrammar)
(key value="0")0/(key)
(key value="1")1/(key)
...
(key value="9")9/(key)
(key value="#")#/(key)
(key value="*")*(key)
</dtlgrammar>
```

[0198] ・flush  
読み取り/書き込み。起動の前に、基礎となる電話インタフェースカードのDTMFバッファを自動的にフラッシュするかどうかを示すブールフラグ。デフォルトは偽になり、タイプヘッドを使用可能にする。

[0199] ・escape  
読み取り/書き込み。DTMF読み取りセッションを終了するエスケープキー。エスケープキーはワシントンである。

[0200] ・numDigits  
読み取り/書き込み。DTMF読み取りセッションを終了させるキーストローク数。エスケープおよび長さの両方を指定した場合は、どちらかの条件を満たすとDTMFセッションが終了される。

[0201] ・dimResult  
読み取り専用ストリング。ユーザが入力したDTMFキーを記述する。タイプした場合はescapeが結果に含まれる。

[0202] ・text  
読み取り専用ストリング。空白で分離されたトークンストリングを記述し、各トークンはDTMF文法に従って変換する。

40 DTMFバッファをフラッシュする。Flushは、DTMFセッション中には呼び出すことができない。

例文: Object. Flush ( ) ;

戻り値: なし

[0207] 4. 3. 3 Flush  
DTMFバッファをフラッシュする。Flushは、DTMFセッション中には呼び出すことができない。

例文: Object. Flush ( ) ;

戻り値: なし

[0208] 4. 4 イベント  
4. 4. 1 onkeypress  
DTMFキーを押すと発生する。これは、HTMLコントロールから継承したデフォルトイベントを上書きする。ユーザがエスケープキーを押すと、onKeyPress ではなくonRecイベントが発生する。例文:

```
50 onKeyPress="speech.Stop(0)">
  <bind targetElement="credit_card_number" />
</DTMF>
<recog id="speech">
  <grammar src="/grm/1033/digits.xml" />
  <bind targetElement="credit_card_number" />
</recog>
```

[0196] 4. 2 属性およびプロパティ  
読み取り専用ストリング。空白で分離されたトークンストリングを記述し、各トークンはDTMF文法に従って変換する。

4. 2. 1 属性  
・dtm(grammar: 必須、DTMF文法のURI。

[0197] 4. 2. 2 プロパティ  
・DTMF(grammar: 読み取りおよび書き込み、ストリング変換行列に対するDTMFを表すXML D 20 ない場合は、電話プラットフォームの内部設定になる。OM ノードオブジェクト (DTMF文法とも呼ぶ)。デフォルト文法は、

[0209] [表15]

インラインHTML	<DTMF onkeypress="handler">
イベントプロパティ	Object.onkeypress = handler
	Object.onkeypress =
	GetRef("handler")

[0210] イベントオブジェクト情報: \* [表16]

[0211] \*

バブル	なし
起動するには	ランタイム電話のキーバンドを押す
デフォルトアクション	押されているキーを送す

[0212] イベントプロパティ: イベントハンドラは ※トは、現在のDTMFオブジェクトを自動的に使用不可プロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。

[0213] 4. 4. 2 onRecog [表17]

DTMFセッションを終了すると発生する。このイベント

インラインHTML	<DTMF onRecog="handler">
イベントプロパティ	Object.onRecog = handler
	Object.onRecog =
	GetRef("handler")

[0215] イベントオブジェクト情報: ★ [表18]

[0216] ★

バブル	なし
起動するには	ユーザがエスケープキーを押す、またはキーストロークの回数が特定の値を満たす
デフォルトアクション	押されているキーを送す

[0217] イベントプロパティ: イベントハンドラは ☆と発生する。このイベントは、認識プロセスを自動的にプロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。

[0218] 4. 4. 3 onTimeout [表19]

タイムアウトまでに、句の終了イベントを受け取らない☆

インラインHTML	<DTMF onTimeout="handler">
イベントプロパティ	Object.onTimeout = handler
(DOMScript)	Object.onTimeout =
	GetRef("handler")

[0220] イベントオブジェクト情報: ◆ [表20]

[0221] ◆

バブル	なし
起動するには	指定のタイムアウト中にDTMFキーストロークが検出されない
デフォルトアクション	なし

[0222] イベントプロパティ: イベントハンドラは ータについてこのイベントオブジェクトに照会を行う可プロパティを直接受け取ることはないが、ハンドラはデ 50 とができる。

【0223】5 CallControlオブジェクト  
電話着ブラウザの電話インタフェース（呼、着来、および接続）を表す。このオブジェクトは、GUIブラウザ中のwindowオブジェクトと同様にネイティブである。したがって、電話オブジェクトの寿命はブラウザインスタンス自体と同じである。電話用の着来ブラウザは、イベントに1つの電話オブジェクトをインスタンス化する。ユーザは、このオブジェクトをインスタンス化または配置しない。  
【0224】この点で、このオブジェクトを通じて、ファーストパーティの呼の制御に関連する機能のみを示す。  
【0225】5.1 プロパティ  
・address  
読み取り専用。XML DOM ノードオブジェクト。実装固有。これは発呼者のアドレスである。PSTNの場合は、ANIとALIの組合せにすることができる。VoIPの場合、これは発呼者のIPアドレスになる。  
【0226】・ringBeforeAnswer  
着信呼に答えるまでの着信音の回数。デフォルトは無限。すなわち、開発者は下記のAnswer（）メソッドを明確に使用して、電話呼に応答しなければならぬ。コールセンタでACDを使用して着信電話呼をキューに入れる場合、この回数は0にセットしてよい。  
【0227】5.2 メソッド  
注：ここに示すメソッドはすべて非同期である。  
【0228】5.2.1 Transfer  
呼を転送する。ブラインド転送の場合、システムは転送が完了すると元の呼を終了し、システムリソースを解放する。  
構文：telephone. Transfer (string, text);  
パラメータ：  
・stringText：必須。意図する受信者のアドレス。  
戻り値：なし  
例外：例えばエンドバーティが話中である、番号が存在しない、ファックスまたは留守番電話が応答するなど、呼の転送が失敗すると例外を投入する。  
【0229】5.2.2 Bridge  
サードパーティへの転送。呼を転送すると、ブラウザはその呼に割り当てられていたリソースを解放することができ、転送した呼がstringIDを使用して戻つてき得る。転送した呼がstringIDを使用している間は、戻つた呼を異なるブラウザに経路指定することができ、呼は、受信者がその呼を終了した際のみ戻ることができる。  
構文：telephone. Bridge (string, string, stringID, [imaxTime]);  
パラメータ：  
・initialTimeOut：任意選択。記録の開始時に許される無音の最大時間（ミリ秒）。  
50

戻り値：なし  
例外：記録をURLに書き込みない限り例外を投入する。  
【0236】5.3.2 onAnswer（）  
着来ブラウザが着信呼に応答すると呼び出される。  
【0237】5.3.3 onHangup（）  
ユーザが電話を切ると呼び出される。このイベントは、プログラムがHangupメソッドまたはTransferメソッドを呼び出して自動的に発生しない。  
【0238】5.4 例  
この例は、電話セッションを操作するために制御脚ペイントに結合（wire）したスクリプティングを示す。  
<HTML>  
<HEAD>  
<TITLE>ログオンページ</TITLE>  
</HEAD>  
<SCRIPT>  
var focus;  
function RunSpeech ()  
if (logon.user.value == "") {  
focus="user";  
p.uid.Start(); g\_login.Start();  
dtmf.Start(); return;  
}  
if (logon.pass.value == "") {  
focus="pin";  
p.pin.Start(); g\_login.Start();  
dtmf.Start(); return;  
}  
p.thank.Start(); logon.submit();  
}  
function login\_reco ()  
res = event.srcElement.recogResult;  
phNode = res.selectSingleNode("//uid");  
if (phNode != null)  
logon.user.value = phNode.xml;  
phNode = res.selectSingleNode("//password");  
if (phNode != null)  
logon.pass.value = phNode.xml;  
}  
function dtmf\_reco ()  
res = event.srcElement.dtmfResult;  
if (focus == "user")  
logon.user.value = res;  
else  
logon.pin.value = res;  
}  
</SCRIPT>  
<SCRIPT for="callControl" event="onIncoming">  
<!--  
// があればアドレスを読み出して、カスタマイズしたものを準備する  
callControl.Answer();  
-->

```

61
//
</SCRIPT>
<SCRIPT for="callControl" event="onOffhook">
<!--
    p_main.Start(); g_login.Start(); dtmf.Start();
    focus="user";
//
</SCRIPT>
<SCRIPT for="window" event="onload">
<!--
    if (logon.user.value != "") {
        p_retry.Start();
        logon.user.value = "";
        logon.pass.value = "";
        checkFields();
    }
//
</SCRIPT>
<BODY>
    <reco id="g_login"
onReco="login_reco0: runSpeech0"
timeout="5000"
onTimeout="p_miss.Start(); RunSpeech0">
    <grammar
src=http://kokaneel/atradedemo/speechonly/login.xml/>
    </reco>
    <dtmf id="dtmf"
escape="#"
onKeyPress="g_login.Stop();"
onReco="dtmf_reco0:RunSpeech0"
interdigitTimeout="5000"
onTimeout="dtmf.Flush();
p_miss.Start();RunSpeech0"/>
    <prompt id="p_main">あなたのユーザIDと個人識別番号を言ってください</prompt>
    <prompt id="p_uid">あなたのユーザIDだけを言ってください</prompt>
    <prompt id="p_pin">あなたの個人識別番号だけを言ってください</prompt>
    <prompt id="p_miss">申し訳ありませんが、聞き取れませんでした。</promp
t>
    <prompt id="p_thank">ありがとうございます。あなたの識別を確認する間お
待ちください</prompt>
    <prompt id="p_retry">申し訳ありませんが、あなたのユーザIDと個人識別
番号が一致しません</prompt>
    <H2>login</H2>
    <form id="login">
        UID: <input name="user" type="text"
onChange="runSpeech0"/>
        PIN: <input name="pass" type="password"
onChange="RunSpeech0"/>
    </form>

```

63

64

```

</BODY>
</HTML>
【0239】6 ダイアログフローの制脚
6.1 HTMLおよびスクリプトを使用してダイアロ
グフローを実装する
次の例は、入力ボックスの値を探して、入力に対して状
況依存型のヘルプを提供する単純なダイアログフローの
<html>
    <title>状況感知型ヘルプ</title>
</html>
<head>
    <script>
        var focus;
        function RunSpeech0 {
            if (trade.stock.value == "") {
                focus="trade.stock";
                p_stock.Start();
                return;
            }
            if (trade.op.value == "") {
                focus="trade.op";
                p_op.Start();
                return;
            }
            //... repeat above for all fields
            trade.submit();
        }
        function handle0 {
            res = event.srcElement.refocused;
            if (res.text == "help") {
                text = "~だけを言ってください";
                text += document.all[focus].title;
                p_help.Start(text);
            } else {
                // proceed with value assignments
            }
        }
    </script>
</head>
<body>
    <prompt id="p_help" onComplete="checkFields0"/>
    <prompt id="p_stock"
onComplete="g_stock.Start0">株式銘柄を言ってください</prompt>
    <prompt id="p_op" onComplete="g_op.Start0">売りまたは買いのどちら
をご希望ですか</prompt>
    <prompt id="p_quantity"
onComplete="g_quantity.Start0">株式数はいくつですか</prompt>
    <prompt id="p_price"
onComplete="g_price.Start0">価格はいくらか</prompt>
    <reco id="g_stock" onReco="handle0: checkFields0">
    <grammar src="/g_stock.xml"/>
    </reco>

```

66

67

68

```

65 <rec id="g_op" onRec="handle0; checkFields0" />
    <grammar src="/g_op.xml" />
66 </rec>
    <rec id="g_quantity" onRec="handle0; checkFields0"
67 />
        <grammar src="/g_quant.xml" />
68 </rec>
    <form id="trade">
        <input name="stock" title="stock name" />
        <select name="op" title="buy or sell">
            <option value="buy" />
            <option value="sell" />
        </select>
        <input name="quantity" title="number of shares"
69 />
        <input name="price" title="price" />
        </form>
        </body>
        </html>

```

【0240】6. 2 SMLを使用する  
次の例は、SML機構を使用したプロンプトおよびeco要素の起動を示す。

```

<html xmlns:it="urn:schemas-microsoft-com:time"
    xmlns:sp="urn:schemas-microsoft-com:speech">
<head>
<style>
    .time { behavior: url(#default#time2); }
</style>
</head>
<body>
    <input name="textBoxOrigin" type="text" />
    <input name="textBoxDest" type="text" />

    <sp:prompt class="time" t:begin="0">
        出発地と行先地を教えてください
    </sp:prompt>
    <t:par t:begin="time.end"
        t:repeatCount="indefinitely"
        <sp:rec class="time">
            <grammar src="/city.xml" />
            <bind targetElement="textBoxOrigin"
                value="//origin_city" />
            <bind targetElement="textBoxDest"
                test="/sml/dest_city[confidence $gt$ 40]"
                value="//dest_city" />
        </sp:rec>
    </t:par>

```

(35)

【0241】7. SMEX (メッセージ) 要素/オブジェクト  
SMEXは、Simple Messaging Exchange/Extensionの略語であるが、これは、クライアントデバイスのプラットフォーム上の外れコンポーネントまたはアプリケーションと通信するオブジェクトである。これは、タグ名<smex>を有する要素として、XMLまたはそれに類似のマークアップベースのドキュメント中に埋め込むことができる。このメッセージングオブジェクトの使用例には、ロギングおよび電話制御を含むことができる。このオブジェクトは、メッセージを通じて新しい機能を追加すること、を可能にすることから、マークアップベースの認識およびプロンプティング (prompting) の拡張性を表す。

【0242】インスタンスを生成すると、このオブジェクトは、その構成パラメータまたは属性指定を通じて、プラットフォームコンポーネントまたはアプリケーションとの非同期的メッセージ交換経路を確立するように指示を受ける。このオブジェクトはストリングプロパティを有し、そのプロパティが割り当て動作 (すなわちvalue) を受ける対象である場合には、必ずその内容がプラットフォームコンポーネントまたはアプリケーションに送られる。同様に、このオブジェクトは、プラットフォームコンポーネントまたはアプリケーションから受け取ったメッセージを保持する、XML DOMノードタイプのプロパティも有する。このメッセージオブジェクトは、プラットフォームメッセージを受け取る必要イベントを送る。このオブジェクトは、その基本動作\*  
例1: ロギングオブジェクトとしてのsmexの使用

```

<smex_id="logServer">
    <param name="d:server"
        xmlns:d="urn:microsoft-com:COM">
        <d:protocol>COMK</d:protocol>
        <d:clsid>2093093029302930293098432098</d:clsid>
        <d:iid>0903859304903498530985309094803</d:iid>
    </param>
</smex>
<listen...>
    ...//resultをカプセルにバインドする他のディレクティブ
    ..... <bind targetElement="logServer"
        targetAttribute="sent"
        value="*[@log SgeS 3]/>
    </listen>

```

【0246】この例は、COMオブジェクトをそのクラスIDおよびインタフェースIDとともに使用して、ロギング機構を実現する仕組みを示している。音声開発者

\*が非同期的なで、アプリケーション開発者がタイムアウト設定を操作するための内部クロックも有する。

【0243】メッセージまたはsmexオブジェクトは、通信手段によってアノノティック (agnostic) である。しかし、一実施形態では、smexオブジェクトは、通常のXMLやマークアップ要素と同じ寿命を有する。すなわち、smexオブジェクトは、それをホストするドキュメントをアンロードすると消滅する。多くのケースでは、smexオブジェクトはアンロードされると自動クリーンアップを実行し、通信リソースを解放することができるが、マークアップベースの永続的な通信リンクが望ましい使用事例 (例えば呼制御など) もありうる。そのような事例のために、このアプリケーションでは、割り振られたリソースを解放する (例えばソケットを閉じるなど) 責任をアプリケーション開発者に置く。

【0244】smexオブジェクトは、メッセージのフォーマット (スキーマ) についてはニュートラルである。実施形態によっては、既存の標準的なメッセージフォーマット (例えばSTPまたはCXML) で使用するものなどをまず優先して、実装者がいくつかの基本的スキーマをサポートすることを必要とするのが望ましい場合もある。基本的に、このアーキテクチャは、プラットフォーム開発者およびアプリケーション開発者の両者が、XMLあるいはそれに類似のマークアップの規格化された拡張性を最大限に活用して、一方では相互操作性を失うことなく他の機能を導入することを可能にする。

【0245】



プを使用することにより、3を超えるか、または3に等しいログ値を有するノードすべてにロギングがすることを示している。この例は、ダウンスレベルブラウザでもアップレベルブラウザでも機能する。  
 【0247】この例はまた、sme xオブジェクトがプラットフォームメッセージを登録ドキュメントに伝達する役割を負うような混乱状態がない限り、あるページが、同じプラットフォームコンポーネントと通信する例＊

例2：着信呼のアドレスの読み取り

```

<input type="text" id="remote"/>
<input type="text" id="transfer"/>
<input type="text" id="local"/>
<input type="hidden" id="session_id" value="session_id"/>
.....
<smex id="telephone" sent="start_listening">
  <param name="server" http://tel-svr/whatever/>
  <bind targetElement="session_id" value="//sid"/>
  <bind targetElement="remote" value="//remote_addr"/>
  <bind targetElement="transfer" value="//transfer_addr"/>
  <bind targetElement="local" value="//local_addr"/>
  ....
</smex>

```

【0249】この例は、どのようにバインドディレクティブを使用し、受信メッセージを処理できるかを示している。この例では、着信呼のメッセージが、下位要素のremote\_addr、transfer\_addr、local\_addr、およびlocal\_addrを有するものと想定しており、その内容はそれぞれ着信呼のリモートアドレス、転送アドレス、およびローカルアドレスを表す。  
 【0250】この例では、HTTPに基づくコネクションスプログミングを使用して電話サーバと通信する。この場合の電話サーバは、複数のブラウザデザインスタンスと通信するように設計されており、したがって、各クライアントは、アプリケーションの開始時にサーバから割り当てられる一意のIDによって自らを識別しなければならぬ。この例では、これはサーバに「start\_listening」メッセージを送信することに

よって実現する。この例では、セッションIDを感じフィードに記憶し、それをウェブサーバに送信して、アプリケーションの次のページをトリガするまでの時間を表すミリ秒単位の数。クロックは、このプロパティに正の値が割り当てられると即時を開始する。この値は、カウントダウンの進行中に変更することができる。ゼロまたは負の値にすると、タイムアウトイベントをトリガせずにクロックを停止する。デフォルトは0、すなわちタイムアウトなしである。

【0251】7.1 プロパティ  
 【0255】status:読み取り専用、オブジェ

クトの最近のステータスを表す整数。可能な値は、0、1、および2であり、それぞれ、正常、タイムアウトの終了、およびプラットフォームとの通信を確立できない、あるいは通信の中断を意味する。受信されるプロパティを通じて、プラットフォーム固有のエラーメッセージを伝達する。エラーメッセージの伝達が成功した場合、ステータスコードは0になる。

【0256】7.2 イベント  
 このオブジェクトは以下のイベントを有する。  
 onReceive: このイベントは、プラットフォームメッセージが到着すると送られる。バインド要素によって宣言されたディレクティブがある場合には、このイベントを発生させる前にそのディレクティブを先に評価する。イベントを送る前に、受け取ったプロパティを更新する。

【0257】onError: このイベントは、タイムアウトが経過したとき、あるいは通信リンクエラーに遭遇したときに送られる。このイベントを送る際、上記のように、ステータスプロパティをそれに対応するエラーコードによって更新する。

【0258】7.3 子要素  
 ある要素の形を仮定するとき、sme xは以下の子要素を有することができる。  
 bind: ディレクティブを受信メッセージに作用させる点を除いては、recoの場合と同様。

```

<smex id="logServer" onload="addFunction()">
</smex>
<script>
  function my_logMessage(logClass, message) {
    logServer.sent = logClass + "|" + message;
  }
  function addFunction() {
    logServer.prototype.logMessage=
    my_logMessage;
  }
</script>

```

よりオブジェクト指向的な方式でこの関数を参照することができ。logServer.logMessage(RECO\_LOG\_ERROR, "My message");上記の例のように拡張を機能させるために、sme xオブジェクトの実装者にはより多くの作業が要求されるが、すべての必要な機能はすでに確立された段階であることに留意されたい。

【0261】  
 【発明の効果】以上、説明したように、本発明によれば、インターネットなどのサーバ/クライアントアーキテクチャで入力の認識を提供するのに使用されるウェブ対応認識アーキテクチャを有するシステムは、統一したアーキテクチャを備えることが可能となる。

【図面の簡単な説明】

【図1】本発明の実施形態の、コンピューティングデバ

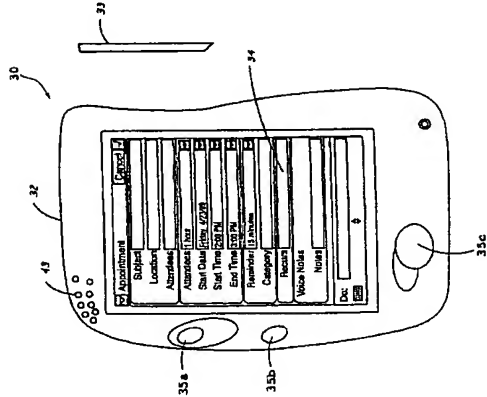
マークアップ言語の例示的ページの図である。  
【図9】本発明の実施形態の、音声レンダリングのみを用い、システム主導型で、クライアントで実行できるマークアップ言語の例示的ページの図である。  
【図10】本発明の実施形態の、音声レンダリングのみを用い、システム主導型で、クライアントで実行できるマークアップ言語の例示的ページの図である。  
【図11】本発明の実施形態の、音声レンダリングのみを用い、混合主導型で、クライアントで実行できるマークアップ言語の例示的ページの図である。  
【図12】本発明の実施形態の、音声レンダリングのみを用い、混合主導型で、クライアントで実行できるマークアップ言語の例示的ページの図である。  
【図13】本発明の実施形態の、サーバサイドのプラグインモジュールによって実行することのできる例示的スク립トの図である。  
【図14】本発明の実施形態の、認識サーバの第1の動作モードを図式的に示す図である。  
【図15】本発明の実施形態の、認識サーバの第2の動作モードを図式的に示す図である。  
【図16】本発明の実施形態の、認識サーバの第3の動作モードを図式的に示す図である。  
【図17】本発明の実施形態の、スク립ティングを用いないクライアントで実行することのできる宣言的マークアップ言語の例示的ページの図である。  
【図18】本発明の実施形態の、スク립ティングを用いないクライアントで実行することのできる宣言的マークアップ言語の例示的ページの図である。

【符号の説明】

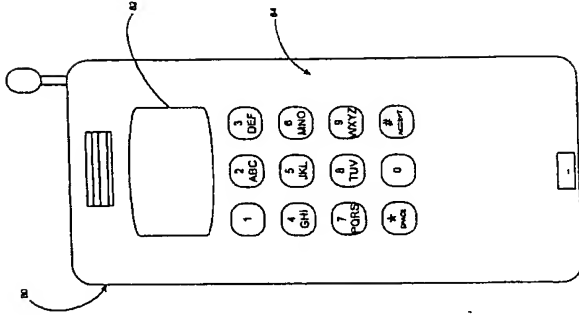
- 29、183 マイクロフォン  
30 データ管理デバイス (モバイルデバイス、クライアント)  
32 筐体  
33 スタイルス  
34 ディスプレイ  
35a、35b、35c ボタン  
36 キーパッド  
37、59 A/D変換器  
43、187 スピーカ  
50 CPU  
52 無線トランスシーバ  
54、152 RAM  
58、151 ROM  
60 通信インタフェース  
80 電話機  
82 ディスプレイ  
84 キーパッド  
120 汎用コンピュータ  
140 プロセッサ  
141 システムバス

307 音声変換システム  
309 パーサ

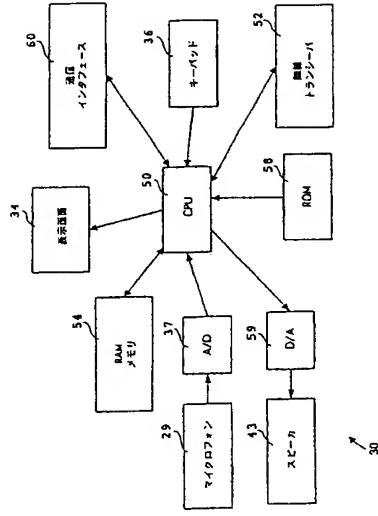
【図1】



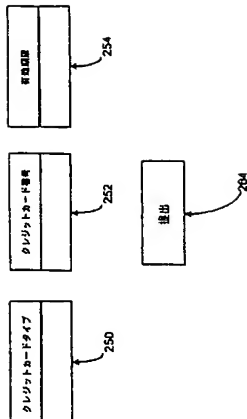
【図3】



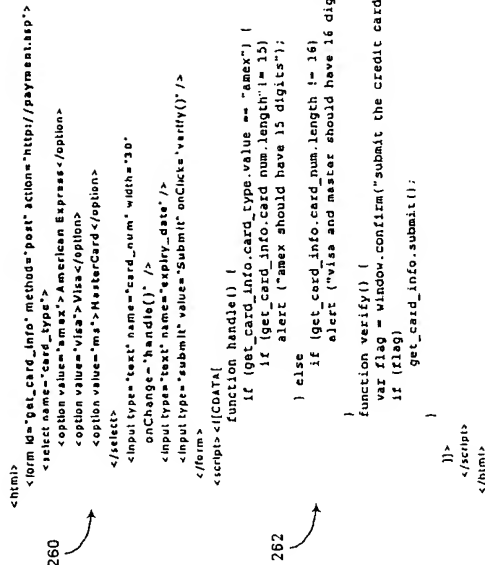
【図2】



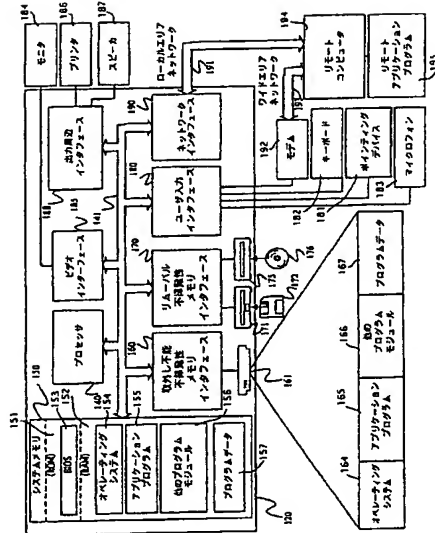
【図6】



【図7】



【図4】



【図5】

